580 / INFORMATION SYSTEMS DIVISION

Implementation Guide for the Use of the Internet Protocol Suite in Space Mission Communications

Release 1.0

July 2004

National Aeronautics and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

Implementation Guide for the Use of the Internet Protocol Suite in Space Mission Communications

July 2004

Prepared by:

____

Jim Rash Date
Team Lead, Code 588

Approved by:

____

Joe Hennessy Date
Chief, Information Systems Division

Information Systems Division
NASA Goddard Space Flight Center
Greenbelt, Maryland

Preface

This communications infrastructure is based on many universally accepted Internet Protocols (IP) supported and used by governments, industry, and academic communities worldwide.

It is not unnatural to envision extending the Internet and Internet communications protocols to Space (Space IP) in support of space missions. The extended communications network would encompass earth and space Internet nodes, including spacecraft, as well as instrument nodes and subsystems within a spacecraft, all of which could be interconnected by communications links carrying traffic conforming to the Internet protocols.

Adapting mission design approaches to realize this vision, and devising methods for incorporating the relevant technologies and standards into missions become central. As always, most of the work required to realize the vision will have to be done in the trenches, mission by mission. However, results from work already done can be documented and made available.

This document provides information to assist in making the necessary decisions concerning the selection and embedding of Internet Protocols and technologies in mission designs to meet specific project requirements.

This document is under control of the Configuration Control Board specified on the signature page. Proposed changes to this document should be submitted to the document owner, along with supporting material justifying the proposed change(s). Changes to this document will be made by document change notice (DCN) or complete revision. Change bars will highlight any revisions from the previous version of this document.

Questions concerning this document and proposed changes should be addressed to the following:

Jim Rash, Code 588
Information Systems Division
Goddard Space Flight Center
Greenbelt, Maryland 20771
USA

Change Information Page

List of Effective Pages	
Page Number	**Issue**
All	Original Release

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
NA	Original	July 2004	NA

Contents

Figures

Tables

SECTION 1 OVERVIEW

1.1 Purpose

Some missions have already adapted the Space IP approach (described in a later section). While the full range of mission operations concepts exceeds what has been implemented to date, this range extends from the relatively simple mission (e.g., CHIPSat) to missions with multiple networked spacecraft and ground assets.

It is the ultimate goal of the Space IP approach to allow a Principal Investigator (PI) and the flight operations team (FOT) to interface with the spacecraft and instrument suite securely, seamlessly, efficiently, and cost effectively throughout the development, integration and testing, and operations phases of the project, using a methodology common to the present worldwide Internet. Although most of the discussion will center on space flight projects, the information contained within this document may be applied towards any type of NASA related activity, including but not limited to balloon, rocket, satellite and other flights.

1.2 Scope

The intent of this document is to provide useful and necessary guidance to the mission project's lead system and communications engineers to enable them to perform trade studies and analyses required to select an appropriate IP mission architecture to meet a set of mission requirements. This document focuses on the potential uses of IP, describes recommended trade studies and architectural concepts for IP missions, discusses various issues relative to using IP, and presents lessons learned from using IP in demonstrations and in actual missions

For information concerning the use of the CCSDS Link–layer framing with IP, please consult the CCSDS Recommendations and Reports for the Advanced Orbiting Systems (AOS) as defined by CCSDS 701.0–B–3, "*Advanced Orbiting Systems, Network and Data Links: Architectural Specifications*", Blue Book Issue 3, dated June 2001.

1.3 Reference Materials

There are several reference materials that should be considered as support "readings" to assist the SE in understanding the use of IP–in–Space; Table 1–1 provides a brief reference list for web references to these materials.

Table 1–1. Reference Information

Web Reference	Web Specifics
http://www.ietf.org/	The Internet Engineering Task Force web site
http://www.freesoft.org/CIE/index.htm	Web–accessible Internet Encyclopedia Reference
http://www.computer.org/internet/	IEEE Computer Society Internet Computing Reference
http://www.computer.org/internet/v2n1/perkins.htm	Mobile Networking Through Mobile IP Tutorial
http://eitsb.gsfc.nasa.gov/docs/ip-in-space.stm	NASA Security Documents for IP–in–Space
http://ipinspace.gsfc.nasa.gov/	Presentations and Documentation on the use of and details for the use of IP for space communications.
http://scp.grc.nasa.gov/siw/	Third in the series of workshops that continued comprehensive discussions of cost–effective technical solutions for the design and engineering of mission communications for deploying Internet technologies for flight missions and identify critical technology development areas.

Most of the documents are part of the Internet Engineering Task Force (IETF) reference materials on the use of IP in general, and some relate specifically to TCP and UDP. These Request for Comments (RFC) related documents are referenced in Appendix G, Request for Comments (RFC) documents.

In addition to that set of documents, there are also several other reference materials that provide more details and specifics on the concepts discussed in this document. The table provides several web sites that were used as reference materials while creating this document:

1.4 Recommendations

This document makes these recommendations for using a complete IP–based architecture.

- The spacecraft should include a general purpose Operating System (OS) functionality
- The on–board system should include a standard IP Application Programming Interface (API), which should at minimum include socket interfaces

1.5 Assumptions

With the recommendations listed in the previous section, this document assumes that future spacecraft systems are implemented using a modern operating system that, at minimum, includes:

- Internet Engineering Task Force (IETF) compliant IP stack. This assumption provides the industry standard IP interfaces for transport, network, and application layer interfaces.
- Layer 2 (Data Link Layer) support; e.g., such as a Frame Relay/High–Level Data Link Control (HDLC) frames.
- File management system, if a file format concept is envisioned for commands and/or telemetry storage or for uplinking and downlinking data files.

1.6 Document Overview

This document is divided into the following sections:

- Section 1 provides the handbook overview.
- Section 2 provides a "lessons–learned" from prior missions that have been launched (either currently operational or completed) or are in development and have used (or are planning to use) an IP approach.
- Section 3 provides architecture overview information to the SE and identifies what trade studies the SE should perform when implementing a mission using IP; this includes both the space and ground segments.
- Section 4 identifies several "operational scenarios" for data transfer when using IP; this includes concepts for space–to–ground, ground, and space–to–space data transfer and what trade studies the SE should keep in mind when implementing the mission using IP.
- Section 5 provides a summary concept of several IP–in–Space security features, which the SE must tailor to ensure that the science and mission data, services, and systems are not compromised.

SECTION 2 LESSONS–LEARNED FROM MISSIONS USING IP

2.1 Missions Using Standard IP Communication Systems

The primary reason for using Internet Protocols on space-to-ground communications links is to take advantage of the hardware and software available in the commercial network world. However, the space environment does pose some challenges that require selecting the proper technologies and protocols for space communication systems. The initial challenge is to identify a data link framing mechanism that supports IP while also performing well over RF links used by space missions.

Some missions have flown, and others are being designed, using a full range of IP technologies in their communication systems. All of these missions use high–level data link control (HDLC) framing, which has been used on over 70 spacecraft during the last 20 years. Most of the spacecraft using HDLC framing were built by the amateur radio community or other low–budget organizations. Many of these missions were developed during the late 1980s and early 1990s when Internet technologies were not as widely deployed. Since many of the missions came from the amateur radio community, they used the amateur radio X.25 (AX.25) protocol over HDLC frames instead of IP.

The NASA community looked at the X.25 protocol in the late 1980s and concluded that it was not suitable for NASA's space communication needs. In the early 1990s, the commercial network world decided that the continual acknowledgement traffic of X.25 was too complex and added too much overhead and potential traffic delay. At this point the Frame Relay protocol was developed as a simple replacement for X.25. As its name suggests, it provides a simple frame forwarding service with no retransmission features and minimal flow control signaling.

The frame forwarding features of the frame relay protocol provide a simple framing mechanism very similar to those traditionally used for space missions (e.g., TDM frames, CCSDS frames). Two major differences from traditional space framing are:

- The combination of frame relay over HDLC provides variable length frames. This allows the frame to fit various size user packets and avoids the packet insertion and extraction processing required with fixed length frames.
- The frame relay protocol has been widely implemented in millions of commercial routers and frame relay switches worldwide. This provides low cost equipment and direct interfaces of space data with worldwide common carrier networks.

The IETF has also defined a standard mapping of IP packets over frame relay and this is widely supported by standard routers.

In 1997, NASA/GSFC started using standard Internet protocols over HDLC on the South Pole TDRSS Relay (SPTR) system. This system uses standard routers at the South Pole and at White Sands to deliver Internet traffic. This connectivity is used for both data and phone service to the South Pole. This service has been and continues to be very successful in providing communication to the South Pole facilities.

In late 1999 NASA/GSFC initiated a search to find a spacecraft that could be used to test the operation of Internet protocols to an orbiting spacecraft. This search identified the UoSAT–12 spacecraft that had been launched in April 1999 as a possible candidate. The primary qualification was that UoSAT–12 used the AX.25 protocol, which meant that it had HDLC framing support in its onboard hardware. It also had a 386 processor, which provided an easier target to find existing IP software.

2.2 UoSAT–12 (Surrey Satellite Technology, Ltd.)

Surrey Satellite Technologies, Ltd (SSTL) designed and built the UoSAT–12 spacecraft which provided a suitable platform for initial tests of IP communication to an orbiting spacecraft. It supported data rates of 38.4 Kbps down and 9.6 Kbps up, which provided an environment similar to dial–up modems. The link did not use any forward–error–correction coding (e.g., convolutional, Reed–Solomon). Consequently the link had to endure some noise and errors but this did not cause any serious problem since the Internet protocols can operate over noisy links. The spacecraft used a DOS–like operating system called the spacecraft operating system (SCOS). Software was developed to incorporate a Berkeley Software Distribution (BSD) IP stack with the operating system. Applications were also developed to support Network Time Protocol (NTP), File Transfer Protocol (FTP), Hyper Text Transfer Protocol (HTTP), and UDP packet delivery applications.

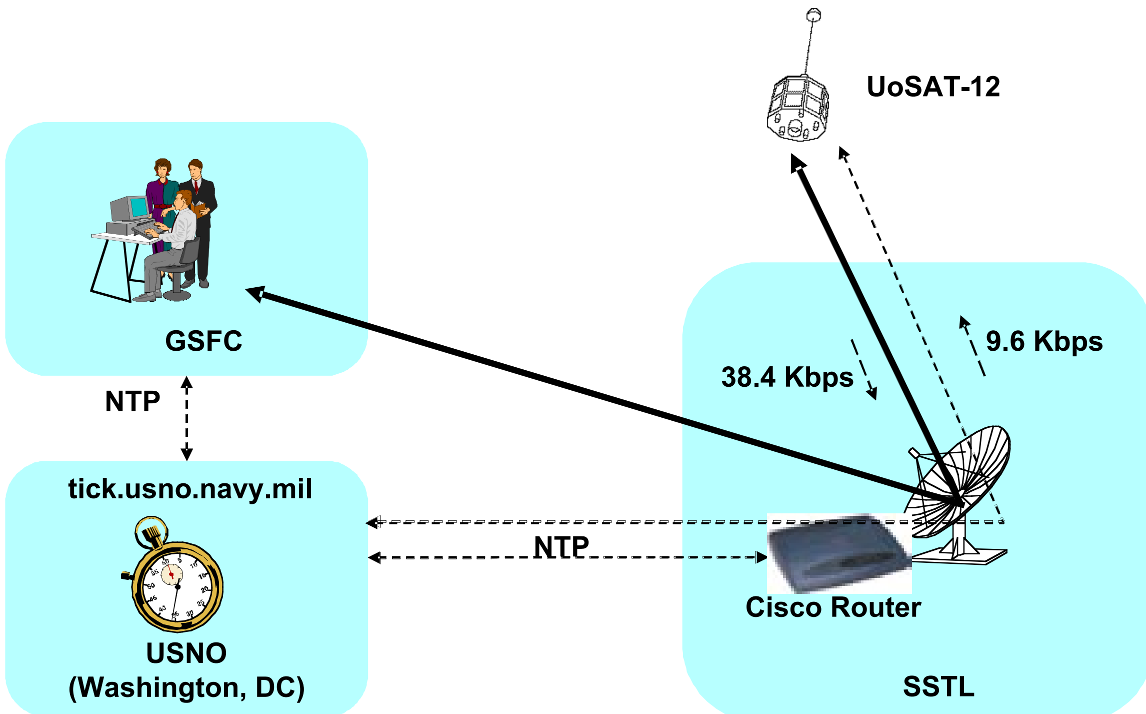

Figure 2–1. UoSAT–12 Technology Overview

The first goal of the UoSAT-12 test consisted of simply sending Ping packets to verify the proper operation of the IP/Frame Relay/HDLC space link communication path. Initial Pings were sent from the router in the SSTL control center directly to the spacecraft. This verified the proper operation of the IP stack onboard UoSAT–12, the RF components, and the ground router. Next, Pings were sent from NASA/GSFC to the spacecraft via the Internet. This demonstrated that the spacecraft was able to perform standard IP addressing functions of accepting IP packets from anywhere and using the sender's address to tell it how to send data back to wherever it came from. Later tests included simultaneous Pings from pre-determined sites in England, California, Pennsylvania, and Maryland. This demonstrated full addressing capabilities with the spacecraft communicating with multiple ground systems simultaneously.

With IP connectivity established, an NTP application was added to the flight software to test the use of NTP for automatically maintaining the spacecraft clock. NTP was set to communicate over the space link to SSTL and then across the ocean to a NTP server at the United States Naval Observatory (USNO) in Washington, DC. This was not an ideal operational environment but it provided a worst–case scenario. Initially NTP was operated in a shadow mode where it did its time synchronization computations but did not actually change the spacecraft clock. After the first passes demonstrated proper operation of the on-board client, some additional passes were completed where

NTP was allowed to check with the Naval Observatory, compute the current time offset, and set the spacecraft clock to the current time. NTP was set to do four time checks every 30 seconds. If it received four consistent measurements, it readjusted the clock. Otherwise, it did nothing and tried again 30 seconds later. On one pass it adjusted the clock by about 320 milliseconds at the start of the pass and then made adjustments of 1 to 5 milliseconds during the pass. At one point near the end of the pass the SSTL controllers intentionally commanded the clock off by over 2 seconds. NTP proceeded to reset it to the correct time at its next execution. This successfully demonstrated NTP operation in space, but further work is needed to determine the exact precision achievable and the effects of CPU speed, link data rate, Doppler, and link errors.

The next test was to install an FTP server onboard and use it to pass files to and from the spacecraft. During one test, two different workstations were used at NASA/GSFC to log into UoSAT–12 using FTP, examine the directory, download files, and upload files. The file transfers completed successfully and the files were received intact. This demonstrated the automated retransmission features of FTP using TCP to provide in–sequence, reliable delivery of data over a noisy link. Network analyzers were used to verify the automatic retransmission of packets to fill in lost packets. This test demonstrated multiple users simultaneously and seamlessly accessing data on the spacecraft.

Another standard type of spacecraft data is real–time data containing spacecraft housekeeping information and science instrument telemetry. To test this type of data flow, an application was added to the flight software that collected parameters onboard, inserted them into a UDP packet, and sent the UDP packets to a specified network address. These packets were received by a workstation running the Integrated Test and Operations System (ITOS) control center software, where the packets were decoded and the individual telemetry values displayed and plotted. These tests demonstrated the ability to use standard UDP/IP/HDLC as an alternative to traditional real–time housekeeping and telemetry using TDM or CCSDS frames.

After the success of all the other protocols, a simple web server supporting HTTP was added to the flight software. Final tests were performed using standard web browsers to access data from UoSAT–12. A simple starting web page was installed on the spacecraft that pointed to some images and a simple telemetry page. The images were retrieved with standard web page clicks. Selecting the telemetry page retrieved a page with some formatted telemetry information, which automatically updated every 10 seconds. These tests proved that standard web server and web browser technologies could be used to access spacecraft data. This is not necessarily an approach that would be used for full operational access to spacecraft data but it might be used as a simple mechanism that anyone could use to provide basic access during integration and test or some operational scenarios.

The final tests with UoSAT–12 occurred during the first Space Internet Workshop at NASA/GSFC. About a week before the workshop, the team considered having a live demonstration of IP data from a spacecraft. The contact times of UoSAT–12 over England were examined but it turned out that all contacts occurred in the evening and very early morning when the workshop was not in session. The best spacecraft contact times to support the demonstration occurred over the west coast of the US, so all that was needed was a supporting ground station. Stanford University had extensive experience working with other spacecraft using the same frequencies as UoSAT–12 and they agreed to provide support. A loaner router was shipped to Stanford on Wednesday, five days before the start of the workshop. Stanford engineers connected the router to the output from their receiver in two days and by Saturday the UDP data packets from UoSAT–12 were flowing to GSFC via the Stanford ground station. Stanford did not have the correct uplink equipment, so packets addressed to the UoSAT–12 spacecraft would be routed to SSTL and not to Stanford. This meant that Stanford only supported a one–way downlink; the UDP housekeeping packets received at Stanford were transmitted to GSFC. This activity demonstrated the ease of installing IP capabilities in a ground station and the ability of UDP packets to find their way to their addressed destination no matter where they are received on the ground.

2.2.1 UoSAT–12 Summary

UoSAT–12 was the first known test of using standard Internet protocols to an orbiting spacecraft using standard off–the–shelf routers and commercial link layer protocols end–to–end. It successfully demonstrated the use of many standard Internet protocols and applications providing a wide range of data delivery options and automated spacecraft operations.

However, the UoSAT-12 tests did not cover all aspects of a full operational mission. It used a simple addressing mechanism where the spacecraft IP address was selected from the subnet at Surrey, which was the only ground station used. This was a normal routable address on the Internet, which meant that any packets addressed to the spacecraft would be routed to Surrey and then forwarded to the spacecraft. This approach worked fine for these tests but does not scale well to missions using multiple ground stations. Scaling for the general case will involve Mobile IP technology, which will be discussed in subsequent sections.

These tests did not use extensive security measures. The very first Ping tests were performed with the spacecraft accessible to the Internet. The main security was that it was only accessible during a few selected passes over Surrey, which only lasted for 6-8 minutes each. During following tests, the router at Surrey had access filters configured, which only allowed packets from a few select locations, such as GSFC, to get through.

These tests only used low data rates of 38.4 Kbps down and 9.6 Kbps up. The router was capable of handling Mbps rates but the UoSAT-12 transmitters and receivers were not built for those rates.

Overall these tests were very successful in demonstrating the ease with which Internet protocols could be used to perform basic spacecraft communication functions. Most of the hardware and software used consisted of standard CPUs, operating systems, network applications, network hardware, and the Internet. These tests demonstrated many of the benefits of using Internet technologies for spacecraft communication.

2.3 AlSAT–1 – Surrey Satellite Technology Ltd

SSTL took the lessons learned from the successful UoSAT–12 IP tests and designed their Disaster Monitoring Constellation (DMC) series of five spacecraft to use IP. The first activity was to clean up the implementation of the IP stack in the SCOS operating system to provide a fully functional, standard application–programming interface for TCP and UDP sockets. The earlier version was not fully standard and required more work to port standard IP applications to the SCOS environment. While the IP stack and applications on UoSAT–12 were added after launch, the AlSAT–1 flight software contained both the traditional SSTL AX.25 support and a standard IP stack as part of the initial design. This was easy because both protocol stacks use the same low–level HDLC framing hardware.

SSTL installed multiple ground stations using standard Cisco routers similar to those used with UoSAT–12. The main difference was to use newer, more powerful routers since the AlSAT–1 downlink runs at 8 Mbps instead of the 38.4 Kbps of UoSAT–12. The network address for the spacecraft was also changed to use private address space and security protocols to get to the router at the ground station. This reflects the shift from the simple test environment used for UoSAT–12 and the full operational environment for AlSAT–1.

The following operational questions, or issues, are under investigation; the document owner will update these items when new information is made available.

- Range of applicability of NTP.
- Do the SSTL spacecraft still use a combo of AX.25 for basic support and IP for moving operational data?
- What performance is realized on the 8 Mbps downlinks?

- Not using Mobile IP means the control center needs to keep track of which ground station is being used and to make proper connections to the correct station.
- What performance and operational characteristics are realized with CFDP file transfers?

2.3.1 AlSAT–1 Summary

AlSAT–1's usage of Internet protocols has worked very well. It allowed AlSAT–1 to operate at data rates of 8 Mbps without any special effort to build custom, high–rate front–end communication processing equipment. A very low–cost off–the–shelf router was able to support the data rates and also provide a good range of security protocols to support the necessary security needs of this international mission. Similar IP protocol usage was designed into three additional DMC spacecraft which were all successfully launched in September 2003:

- BILSAT for Turkish customer Tubitak–ODTU Bilten
- NigeriaSat–1 for Nigerian customer National Space Research & Development Agency
- UK-DMC funded by the UK government/BNSC.

The UK-DMC spacecraft also has an experimental version of a Cisco mobile router onboard the spacecraft. This system is expected to undergo tests of Mobile Routing protocols and other advanced Internet protocols in 2004.

2.4 CHIPSat – University of California Berkeley/NASA

The Cosmic Hot Interstellar Plasma Spectrometer Satellite (CHIPSat) is the first NASA mission to utilize the IP/Frame Relay/HDLC protocol stack as its communication system. This is also the first mission known to use IP/Frame Relay/HDLC as its sole communication protocol stack with no alternate communication mechanism. The CHIPSat mission designers closely monitored the tests performed with UoSAT–12 and decided that IP provided the most cost effective solution for CHIPSat. CHIPSat is a NASA University-class mission with a budget cap of $14M for all aspects of the mission. The mission designers were looking for solutions that would provide the maximum capability for the least cost. Using IP allowed them to save significant time and effort in their communication system by simply using existing Internet hardware and software and capabilities built into standard operating system software.

CHIPSat does not use Mobile IP but instead uses private, non-routable address space for the spacecraft address, ground stations, and control center. Virtual Private Network (VPN) tunnels are used to secure connections between the control center and ground stations. CHIPSat can address packets such as UDP housekeeping data directly to the private address at the control center. This data is routed to the control center within the static routes of the CHIPSat VPNs. Many of the other data transfers are performed over the point-to-point link from the current ground station computer to the spacecraft, so no Mobile IP routing is needed.

The spacecraft takes measurements and records them in files along with housekeeping information. During ground contacts, automated scripts use FTP/TCP to retrieve the files from the spacecraft. These transfers occur between the spacecraft and a Linux system at each ground station and the files are later transferred from the ground station computer to the control center at UCB. Files with stored commands are also uploaded to the spacecraft using FTP.

Using the open Internet as part of the control center-to-spacecraft communication path has presented a few problems. It provides a very low cost wide-area network, but it does not provide the highly reliable communication path used by other NASA spacecraft. Any congestion or circuit outages on the Internet can result in loss of connectivity and the CHIPSat operators just have to wait until other organizations resolve those problems. Using UDP packets for basic housekeeping and telemetry means that some packets have gotten lost between the ground stations and the control center. However, this loss has been within allowed limits. As mentioned earlier, most of the file transfers occur initially between the spacecraft and computer at the ground station. This helps avoid some of the issues with delay and congestion on the open Internet.

CHIPSat uses standard NTP to automatically maintain its spacecraft clock. It does not require high precision (e.g., millisecond) spacecraft timing and NTP provides sufficient precision in a fully automated environment, which helps in reducing operational costs.

2.4.1 CHIPSat Summary

The CHIPSat use of Internet protocols as its sole communication mechanism has worked very well. The CHIPSat mission has met all of its objectives and the communication system has performed flawlessly. NTP has provided a simple, automated mechanism to maintain the spacecraft clock. However, CHIPSat does not have any high–precision time resolution requirements and NTP just needs to keep the time to sub–second accuracy to help keep track of file times and simple time stamps.

CHIPSat uses the standard FTP/TCP protocol for its file transfers, which is not highly optimized for space use. However, there is sufficient spacecraft contact time so efficiency is not a key issue.

2.5 CANDOS – NASA GSFC

NASA/GSFC had already performed a wide range of tests using Internet protocols in space, but in late 2001 an opportunity arose to perform more advanced tests as part of the Communication And Navigation Demonstration on Shuttle (CANDOS) mission. A primary goal of this experiment was to test the new Low–Power Transceiver (LPT) in the space shuttle payload bay during the STS–107 mission. The LPT provided a transceiver capable of supporting multiple data rates, using both NASA ground network (GN) stations and space network (SN, TDRSS) relay satellites. The data rates used during the mission ranged from 2 Kbps up to 128 Kbps in various combinations of symmetric and asymmetric rates as well as one–way links. The payload included an x86 processor running a version of Linux to control the transceiver, perform GPS computations and run Internet applications. When launch delays provided additional time to incorporate new concepts, the project decided to install additional software to investigate the performance of Internet protocols and applications to provide more automated operation and increased security.

In the following photo, the LPT is the shiny, shoebox-size object in the lower left just under the round 12-inch high-gain transmit antenna. There are three other lower gain but wider angle 3-inch antennas for low-gain transmit and receive and GPS receive. The components were all mounted on a truss installed in the tail of the shuttle bay.

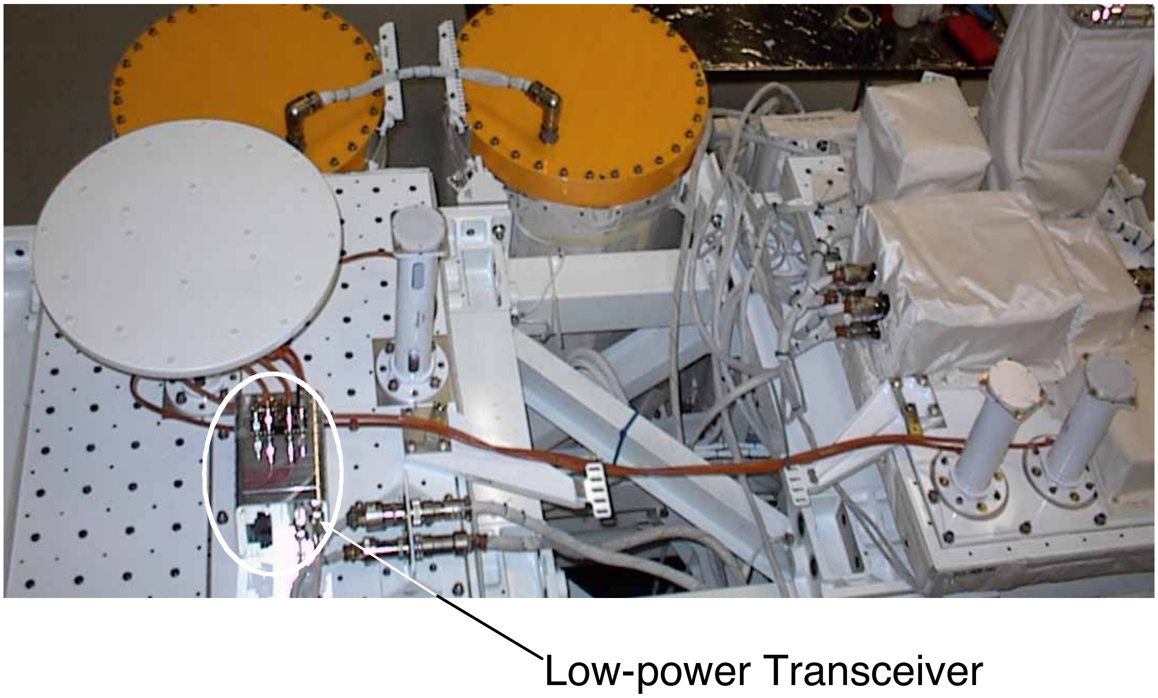

Figure 2-2. CANDOS Photo Identifying the Low–Power Transceiver

Some of the primary Internet Protocol and automation experiments added involved:

- Using Mobile IP to automatically set up routing tunnels to send uplink traffic to the correct GN or SN location for uplink
- Using the Multicast Dissemination Protocol (MDP) for automated, reliable file transfers using UDP
- Running NTP for multiple days and letting it perform clock setting as well as manage clock drift
- Using secure shell (ssh) and secure copy (scp) to perform secure access to the payload

The shuttle launched on schedule Jan. 16, 2003 and approximately four hours after launch the CANDOS payload was powered on. The processor booted and the transceiver was configured in its default mode. A housekeeping process was always running to collect status information and send a status packet to the ground every 10 seconds. These packets were sent as simple tab-delimited ASCII strings in a UDP packet addressed to a computer in the CANDOS control center. There the packets were logged to a file, displayed on a screen, and forwarded to other systems for additional displays. These UDP packets were the first data received from CANDOS and they began arriving as soon as the ground receiver locked up and a downlink was established. No uplink or Mobile IP functionality was required to receive this data. These UDP housekeeping packets provided very useful status information and proved to be a very reliable indication that the downlink was operational. The control center was often able to verify that the downlink was operational before confirmation was reported by the ground station itself.

Once two-way communication was established on the first communication pass, the Mobile IP daemon detected advertisements from the ground station and automatically set up a routing tunnel. This path was used to initiate an FTP transfer to upload software upgrades. However, due to the slow 2 Kbps uplink, the FTP file transfer only completed 90% of the 300 KB file during the contact. It was only 11 minutes to the next scheduled contact, so a decision was made to wait and see if the FTP session would continue. When the next contact began, Mobile IP automatically established a new routing tunnel and after one minute, the FTP session resumed and completed the remainder of the file transfer. This demonstrated the combination of Mobile IP changing a routing path completely transparent to any applications currently active and FTP/TCP automatic retransmissions recovering after a link outage. If the link outage had been much longer than 12-15 minutes the TCP session would have timed out and dropped the session. However, this time range can be adjusted by changing operating system parameters controlling TCP operation.

After the first day of CANDOS operation, the indication of the Mobile IP tunnel being established became the primary indicator that there was a good two-way link to the payload. Mobile IP requires a two-way link for the advertisements to go up and registration requests to come down. However, Mobile IP only needs to get one packet up and one packet down to set up a routing tunnel. There were times when a tunnel would come up but PINGs were only intermittently successful. This simply indicated that a few packets got through to set up the Mobile IP session, but the link still had enough errors to interfere with many PING request/response sequences. The real benefit of Mobile IP was the way it automatically handled IP routing to the current SN or GN station and did it all transparent to the upper layer protocols (TCP, UDP, FTP, NTP, etc.).

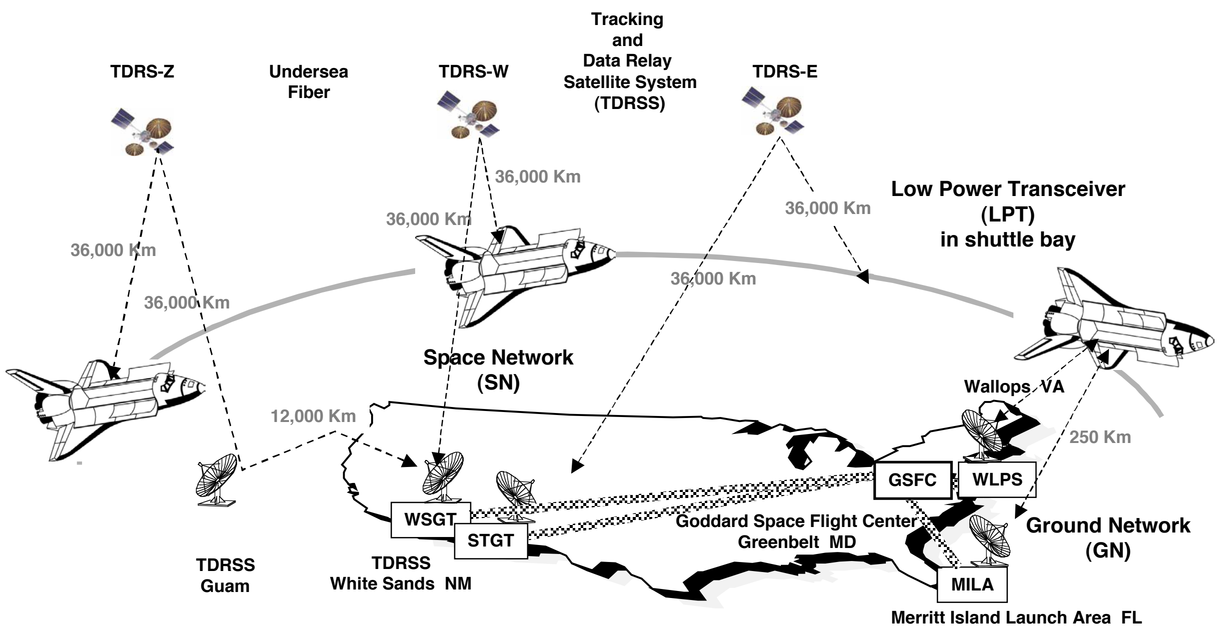

Figure 2–3. Mobile IP Network Connectivity for Shuttle to SN/GN Stations

However, since Mobile IP does require a two-way link to determine the current link, it will not work in spacecraft communication scenarios with one-way links with only an uplink or downlink. These scenarios were examined during the CANDOS mission. Due to TDRSS resource constraints, it is often easier to schedule a TDRSS downlink only, without requiring an uplink. In this mode, packets simply arrive at a ground router and they are routed according to their destination address. This mode was exercised by scheduling TDRSS return-only service with processes onboard CANDOS scheduled to only send UDP traffic and not require any uplink.

The one-way downlink mode also occurred naturally during the initial minute of GN contacts. During GN contacts the spacecraft transmitter was normally on and housekeeping UDP packets were being sent continually. At a GN site the receiver locked up and these housekeeping packets were immediately passed to the control center. The GN site procedure was to send an unmodulated carrier to the spacecraft and the control center monitored the housekeeping data to verify that LPT receiver had locked onto the ground signal. Then the ground station began modulating the uplink and the two-way connection was complete, and then Mobile IP would set up its tunnel.

The other one-way link scenario of an uplink without any downlink is often referred to as "blind commanding". This may be used operationally to command the spacecraft transmitter on, and it is also used in anomaly situations to attempt to blindly send commands to a spacecraft in the hope that it will respond. Often a spacecraft will keep its power hungry transmitter turned off most of the time but keep its receivers turned on. Then the ground can send a "blind command" turning on the transmitter and establishing a two-way link.

Mobile IP becomes necessary when trying to send data to a spacecraft that is moving among many stations. When the control center tries to send a packet to the IP address of the spacecraft, the ground routers do not know where to route the packet. If an old Mobile IP tunnel was still active and had not timed out yet, the packet would be sent to the last station and the Foreign Agent (FA) router would try to uplink it. However, normally the Mobile IP tunnel would have timed out and the Home Agent (HA) router would not know where to send the packet. This is where the Mobile IP protocol normally steps in with the Foreign Agent advertising, the Mobile Node responding, and the Foreign Agent informing the Home Agent where to route packets. With a one-way uplink, this process will not work and another approach must be available to get commands to the spacecraft. On CANDOS, a manual routing approach was demonstrated in which the ground network operators manually configure the Home Agent and the proper Foreign Agent by setting up the tunnel and two associated static routes. This provided exactly the same IP packet routing function of Mobile IP, but it required human intervention to set up the routing instead of having Mobile IP automatically set up the tunnel and routes.

Security protocols incorporated in the CANDOS mission included the SSH and SCP protocols along with Mobile IP security mechanisms. The Mobile IP protocol requires the use of authentication mechanisms between the Mobile Node (e.g., spacecraft) and the Home Agent (e.g., control center router). This prevents unauthorized systems from using the Mobile IP services advertised by the Foreign Agents at ground stations. Security authentication was also turned on between the FAs and the HA to provide further security. Mobile IP was configured with static authentication keys, which provide basic security, but future missions may want to consider security approaches with more dynamic key management mechanisms.

One major test of the LPT was to use its GPS receivers and the GEODE software to compute the location and speed of the shuttle. This experiment generated many multi-megabyte files, which needed to be retrieved during the mission. Other files were also generated onboard to log basic housekeeping information and NTP performance. Most of these data files were transferred to the ground using the Multicast Dissemination Protocol (MDP). The MDP protocol uses the UDP transport protocol instead of TCP. This makes it especially well suited to space use since it is not sensitive to propagation delays and can even function across one-way links. The MDP application supported a "hot directory" that allowed files to be collected there between passes and then automatically transferred to the ground when the link came up. It also allowed the control center to prepare a directory of files for automated uplink once Mobile IP established an uplink route.

Since the LPT processor was running Linux, much of the command and management of the transceiver and applications was performed using standard shell scripts and automated operations initiated using the "cron" process. Many commands consisted of an ASCII string containing the name of a shell script and parameters to pass to it. This allowed the addition of many new commands during the mission by simply uploading new shell scripts and then invoking them in future commands. For a long-term operational mission, this process should be made more secure by encrypting the command string before placing it into the UDP command packet.

During the mission some additional scripts were uploaded to demonstrate how IP enables a payload to send selected data to multiple destinations based on onboard decisions. Some data was sent to one destination, other data to a different destination, and some data was sent to both destinations. This demonstrates the full network addressing capabilities of Internet protocols where the source system or space payload in this case, can determine where it wants various data packets sent to. This is completely different from current spacecraft where the only information provided on packets is the data source. In current TDM and CCSDS mechanisms, the ground data routing must be managed by other mechanisms that normally require extensive scheduling and manual interaction on the ground.

CANDOS also performed tests of the NTP protocol performing long-term clock maintenance. NTP does require a two-way connection so it was not able to update the time during one–way contacts. It did track clock drift and attempted to adjust the onboard clock to adjust for it. It often maintained the clock to within 10–20 milliseconds, but at times it varied beyond that range. More work is needed to determine the limitations of NTP timing accuracy in the space environment.

2.5.1 CANDOS Summary

All mission objectives were met for the CANDOS mission. The mission successfully demonstrated functions such as HDLC framing, IP packet encapsulation over Frame Relay, and Mobile IP. Standard Internet applications such as SSH, SCP, Telnet, FTP, MDP, and NTP operated properly but their performance was a function of the varying uplink/downlink rates (from 2 Kbps to 128 Kbps) used across the contacts.

The UDP/IP/HDLC housekeeping packets and blind commands (using UDP) provided functionality identical to TDM and CCSDS frames currently used by spacecraft. The main advantage is the support for Internet protocols and the ease with which software (operating systems support, PERL scripts, and applications) can be developed to use them.

This mission demonstrated Internet protocols operating in space. However, a long-term operational mission still needs more security solutions for things like dynamic key management, user ID/password management, and more automated onboard file management.

2.6 Additional Missions Lessons Learned

More sections will be added as more lessons learned are documented for each mission that has used IP or is beginning to use IP for some portion of the data transport. The next operational missions to document will be the following SSTL missions which launched in September. 2003.

- BILSAT for Turkish customer Tubitak–ODTU Bilten
- NigeriaSat–1 for Nigerian customer National Space Research & Development Agency
- UK-DMC funded by the UK Government / BNSC

NASA missions that are contemplating using IP or that have baselined IP:

- Global Precipitation Measurement mission
- Magnetosphere Multiscale mission
- International Space Station

Other missions that are contemplating using IP or that have baselined IP:

- Citizen-Explorer (University of Colorado, Boulder, CO)
- EagleEye (Embry-Riddle Aeronautical University, Daytona Beach, FL)
- LionSat (Penn State University, State College, PA)
- MidStar (US Naval Academy, Annapolis, MD)
- NPsat (Naval Postgraduate School, Monterrey, CA)

2.7 IP Mission Lessons Learned Summary

All of these missions have provided a base of knowledge and experience on the use of Internet Protocols for a range of space communication environments. The basic lesson learned is that through an understanding of a mission's communication needs and end-to-end system engineering and design, Internet protocols and technologies can be selected to meet those needs. The following lists traverse the protocol stack upwards to summarize lessons learned from space missions that have used Internet protocols.

Data link Protocols (framing, frame error detection, virtual channels)

- HDLC framing provides a simple framing mechanism that has been used in space communication systems for over 20 years.
- HDLC framing supports variable length frames, which allows simple packet insertion and extraction by putting one packet per frame.
- HDLC framing always uses a CRC-16 error check to identify and discard any frames with bit errors. Any frames received are intact and further data processing is simplified, since data either is good or is not forwarded to the destination.
- HDLC framing is not affected by one-way links or propagation delay, since it has no ACK/NACK mechanism and it operates identically over links at any data rate, distance, or delay.
- HDLC framing does require a clean enough link to receive a frame of data with no errors. If a link has a high error rate, it should be cleaned up with forward error correction (FEC) coding such as convolutional coding, Reed-Solomon, Turbo Product Codes, or Low Density Parity Check. The combination of FEC and HDLC provides excellent data recovery for space links.
- Frame relay headers and RFC 2427 (Multi-protocol Encapsulation over Frame Relay) provide a standard serial link format that is supported by numerous COTS network equipment vendors.
- The Frame Relay data link connection identifier (DLCI) field can be used to provide 1024 virtual channels similar to current CCSDS virtual channels.

Network Protocols (packet addressing, packet routing)

- IP addressing provides fully identified packets with both source and destination addresses, which identify both where the data came from and where it should be delivered. This mechanism enables data-driven data delivery, as opposed to the scheduled processes used for current spacecraft. This allows the spacecraft to control where the data is routed whether it be different facilities or other spacecraft.
- IP addressing is not affected by one-way links or propagation delay, since it has no ACK/NACK mechanism and it will function over any link at any distance.
- The overhead of IP packets is higher than for legacy space protocols but provides more functionality and the overhead is a direct function of packet size. With larger packets of 1000 to 1500 bytes the overhead difference is insignificant (1% to 3%).
- Tools for IP communications trouble shooting are off-the-shelf items.
- Mobile IP provides a lightweight mechanism for the spacecraft and ground station to automatically set up a route for sending packets to a spacecraft without knowing which station is scheduled.
- Mobile IP does require a two-way communication link to operate. If a two-way link is not available, the IP routing must be set up manually in the routers, as might be the case for blind commanding.

- Mobile IP only needs to get three packets across the space link in order to set up a route. It can operate over links that have high error rates (even worse than 10^{-5}).
- PING (ICMP echo request/response) provides a simple, standard mechanism for verifying proper end-to-end operation of a two-way IP data path to a spacecraft.

Transport Protocols (subchannels (ports), unreliable/reliable delivery)

- UDP packets provide an alternative to legacy TDM and CCSDS frame/packet mechanisms for sending packets of data across one-way and two-way data links.
- UDP packets are not affected by propagation delay and will function over any distance.
- UDP packets support 65,535 subchannels (ports) to identify different types of data. This mechanism can be used similar to current CCSDS APIDs.
- The standard UDP socket application programming interface (API) provides a simple, standard mechanism for sending and receiving data using any programming or scripting language.
- TCP provides a mechanism for reliably delivering a byte stream across a two-way communication link, but it requires a two-way link in order to operate.
- TCP performance is a function of data rate, propagation delay, and link error rates.
- TCP can be used without any modifications over space links if the delays are reasonably low (e.g., 1 to 2 seconds), data rates are low (tens of Kbps), and error rates are low ($<10^{-6}$).

Applications

Time measurement --

- PING (ICMP echo request/response) provides a simple, standard mechanism for measuring the round-trip propagation time over a network data path.
- PING combined with the IP header timestamp option provides a standard mechanism for measuring the relationship of a spacecraft clock and ground network devices. This provides a mechanism for reading the spacecraft clock and its relationship to the time of the uplink echo request and the receipt of the echo response at the ground station.
- NTP provides an automated mechanism for maintaining a spacecraft clock to a fraction of a second. The absolute accuracy that NTP can achieve over various space links and satellite computers is a function of many variables and has not yet been determined.

File transfer --

- The multicast dissemination protocol (MDP) provides UDP-based reliable file transfer that performs well over space links with minimal performance degradation due to data rates, propagation delays, and link bandwidth asymmetry.
- MDP supports options for doing highly reliable file transfers over a one-way communication link by sending additional Reed-Solomon forward error correction packets separate from the data file.
- FTP will work for performing reliable file transfers in space communication environments and is being used operationally by the CHIPSat mission. However, since FTP operates over TCP, it should only be used in conditions where the impact of data rates, propagation delays, link bandwidth asymmetry, and link errors have been properly analyzed and understood.
- SCP performs reliable file transfers over TCP just like FTP. It does add some additional overhead for setting up a secure connection and for encrypting the data over the link. Otherwise its performance issues are similar to FTP/TCP.

- CFDP performs reliable file transfers over UDP and performs similar to MDP. It does not support the one-way FEC assisted file transfers or multicast addressing options of MDP.

Remote login --

- Telnet provides a low-overhead, remote-login service that will operate over space links. It allows an operator to directly perform command and control functions to a spacecraft's operating system with reliable data transfer provided by TCP. However, since Telnet operates over TCP, it should only be used in conditions where the impact of data rates, propagation delays, link bandwidth asymmetry, and link errors have been properly analyzed and understood.
- SSH provides a remote-login capability similar to Telnet, but it first establishes a secure connection and then encrypts all data transferred over the link. There is some additional overhead for the secure connection setup and encryption. Otherwise its performance is similar to Telnet/TCP.

Test and Analysis Equipment

- LAN analyzers are readily available to receive, store, and decode IP traffic on Ethernet LANs. These tools range from free, public-domain software (e.g., Ethereal), to commercial software packages, to full hardware systems. These analyzers support full packet decodes for standard IP protocols and some allow users to add decodes for additional protocols.
- WAN analyzers are readily available to receive, store, and decode HDLC frames carrying IP traffic on serial links. These provide insight into the exact data packets traveling across the space link.
- LAN and WAN analyzers provide immediate insight into data packets flowing on communication links without requiring any development of special test equipment.
- LAN and WAN analyzers often identify traffic that is generated automatically by various protocol stacks and that may not be expected by the system designers.
- Decoded packet information such as timestamps, sequence counts, and packet lengths can be extracted from LAN/WAN analyzer decodes and processed with standard spreadsheet packages to understand protocol performance issues.
- Plots of protocol parameters can provide quick analysis of protocol performance and identify periodic events that might be missed during normal examination of the data.
- Protocol analysis software such as "tcptrace" is freely available to perform more complex analysis and plotting of TCP operation by examining packets captured by LAN and WAN analyzers.

While some Internet protocols may not be well suited for particular types of space communication environments, the Internet protocol suite provides a wide enough range of protocols so that with a bit of analysis and understanding, standard protocols can be identified to meet the communication needs of space missions. A significant benefit of being able to select from these standard protocols is their wide support and implementation in operating systems, applications, network equipment, and test equipment. Missions like CHIPSat and the SSTL DMCs (AlSAT-1, NigeriaSat-1, BILSAT, and UK-DMC) have used this to keep costs down while rapidly developing, deploying, and operating new spacecraft.

SECTION 3 ARCHITECTURE

The following sections provide further details concerning the separate mission architecture and developmental phases. These subsections identify the potential tradeoffs or analysis efforts that the systems engineer (SE) would perform in order to efficiently design the overall ground and space segments, including both hardware (HW) and software (SW) elements.

3.1 Architecture Trade Studies

The following are types of trade studies that the SE would perform during the initial architecture phase. While several of these studies (such as forward error correction schemes) would already be on the list of studies that the SE would perform, there are several new trade studies that are IP–related that the SE must begin to understand, identify, and complete during the architecture phase. These studies will drive the follow–on mission requirements and development phases. The following is a "*living* list of these trade studies; this list will be updated as new studies are identified.

- Interfaces with instrument suites and sensor interfaces – single IP address for entire s/c or multiple addressed spacecraft and instrument suite on an on–board Local Area Network (LAN)
- Data throughput and data storage capacities on–board the spacecraft
- Redundant LAN/Routers/Switches, or BUS, architectures
- Use of UDP/IP versus TCP/IP for all spacecraft communications
- TCP–window size, if using TCP/IP (and whether to use standard or large windows) is based upon the on–board memory allocation scheme and the type of computer and OS chosen for the spacecraft. A TCP window is the amount of outstanding (unacknowledged by the recipient) data a sender can send on a particular connection before it gets an acknowledgment back from the receiver that it has gotten some of it.
- File management applications, or simple TCP/UDP APIs; this would dictate whether all data is transferred in real–time, or whether the data is stored for later downlink
- Two–way versus one–way communications between the ground and spacecraft or between spacecraft
- Using legacy spacecraft building concepts versus "state–of–practice" or even "next generation" concepts
- Radiation hardened RS–422, Military Standard 1553 (MIL–STD–1553), HDLC, Digital Video Broadcast (DVB), CCSDS Transfer, Asynchronous Transfer Mode (ATM), or Packet over a Synchronous Optical Network (SONet)
- Radiation hardened Ethernet, Fiber Distributed Data Interface (FDDI), IEEE–1355–Spacewire, or IEEE–1394–Firewire
- IP applications that result in an impact to the bit error rate (BER).
  - IP applications can aid in the overall reduction of the mission requirements for the BER. Using a UDP–based reliable protocol to support the automatic retransmission of data rather than the current approach that requires a stricter BER limit to ensure that data are delivered without error. The reduction in the BER can be used as a tradeoff involving either increased data downlink rates or lower power requirements.
- RF implementation schemes that impact the link performance requirements, such as using smaller, re–configurable transceivers (e.g., Low Power Transceiver (LPT)), cell phone technology, or wireless Ethernets, frequency allocations or reuse, and dynamic power management.

- Build–your–own ground station versus IP Interface modifications to existing radio frequency (RF) equipment at a station
- IP security for both data and systems integrity; what uplink and sensitive downlink data needs to be encrypted

During the architecture design and development phase, the SE leads all of the various groups to architect and develop their part of the complete spacecraft and instrument suites using an IP concept. The mission engineer would maintain the group's focus on the various trade studies as listed above as well as propose the tradeoffs based upon the types of operational scenarios defined in Section 1.

The SE would determine how the data is routed between and among the instruments and the flight software (FSW) subsystems and identify the data transport requirements (redundancy, data throughput and reliability, etc.).

The SE would determine whether the spacecraft should have a single IP address (where data would be routed among the various onboard "tasks" via internal messages) or whether the spacecraft should have multiple IP addresses, one for each on–board component, and determine whether the spacecraft should have an on–board router or equivalent software to support the multiple IP addressing scheme.

The SE would verify whether the spacecraft was required to store science and telemetry data for later downlink or whether the spacecraft would be in continuous station coverage and would downlink all data in real–time during the corresponding station contact. The SE would also determine whether the spacecraft should have any type of file storage requirements to support commanding features. These two trades would identify whether a file management system is required or whether real–time socket connections alone would suffice to transfer telemetry and command data. If a file system is required to support the science requirements, the SE would determine the mechanism used for file maintenance, and would determine how files should be downlinked in a reliable manner to ensure the data completeness requirement.

The SE would coordinate the IP security (IPsec) concepts for this mission. Security solutions should be tailored to an appropriate level for each mission based on factors such as mission size, acceptable level of risk, and mission budget, just to name a few. This is discussed in more detail in Section 5

3.2 Space Segment Architecture Analysis

The SE must determine the on–board routing mechanism to transfer data within the spacecraft. The SE would determine the on–board network requirements to move science and engineering data and to eventually transfer that data to the ground. There are several proposed solutions that support a message routing capability, such as LAN cards, on–board routers and gateways. Table 3–1 provides examples of how IP can support various space missions and also identifies some underlying aspects of what features are required for each identified mission profile. This table is not all encompassing in that it currently does not identify all mission classes or all aspects of IP; however, it is the intent to update this table as new information is developed. This table is meant only to provide initial guidance to the SE; the SE would use this table as a starting point to identify how his mission requirements compared against these examples. The SE would then have a good starting point for determining which IP attributes should be considered as part of the architecture.

Table 3–1. Representative IP Approaches Based on Mission Characteristics

Mission Class	Single IP Address	Multiple IP Address	IP Security	Mobile IP	Dedicated Ground Station (GS)	Multiple GS
LEO, Low Data Rate (< 2Mbps); 8–10 minute ground contacts; C&DH system acts as single point of communications; Supported by Figure 3–1,	X		X	X		X
LEO, High Data Rate (>2Mbps); 8–10 minute ground contacts; C&DH system acts as single point of communications; Hidden IP network on–board s/c Supported by Figure 3–2,	X		X	X		X
LEO, High Data Rate (>2Mbps); 8–10 minute ground contacts; IP network on–board s/c using Multiple IP Addresses Supported by Figure 3–3,		X	X	X		X
GEO, Low Data rate (< 10 Mbps); Long duration ground contacts; C&DH system acts as single point of communications; Supported by Figure 3–1,	X		X		X	
GEO, High Data rate (> 10 Mbps); Long duration ground contacts; IP network on–board s/c using Multiple IP Addresses Supported by Figure 3–2,		X	X	X		X
Deep Space mission; low data rate; large time delays for transmission of cmd & telemetry; Very–long duration ground contacts; Uses DSN and/or 34–m ESA stations Supported by Figures 3–1, 3–2, & 3–3	X	X	X	X		X
Constellation of missions cross talk between satellites		X	X	X		X

3.2.1 IP Concepts for Use On–Board the Spacecraft

This document defines several architectural alternatives reflecting the system engineering analysis of the trade–offs necessary to implement IP. These alternatives can be initially though of as:

- A simple IP implementation approach
- A slightly more complex IP approach
- A more complex approach, which could include Mobile IP/Routing with multiple IP addresses

These architecture levels help the system engineer understand the Internet protocols and concepts and make trade decisions based on what software (SW) and hardware (HW) applications are required to support the use of IP.

3.2.1.1 Simple IP Approach for Spacecraft and Instrument

A basic approach for implementation is for the system engineer to decide that the spacecraft will have only one IP address and that it resides within the command and data handling (C&DH) system. This approach closely mirrors the heritage non–IP approach in which all data is routed to the C&DH system, which acts as a SW router for routing spacecraft and instrument telemetry data to the ground or for routing the command data to the various SW tasks, HW tasks, or instrument components.

For this approach, we would also assume a low earth orbiting (LEO) spacecraft with well–defined ground station contacts at predetermined sites. Standard IP connectivity concepts would be used over predefined wide and local area networks (WANs and LANs). The SE would perform trade studies to determine the requirements for data reliability and transfer mechanisms and whether communications are routed as UDP–based or as TCP–based messages. One advantage to this approach is that current legacy concepts can still be used on–board the spacecraft in terms of using a MIL–STD bus architecture and a high–speed serial data interface to route data on–board the spacecraft; the MIL–STD bus is used for standard communications with HW sensors and other generic HW devices while a high–speed serial interface, like RS–422, would be used for data transfer with science instruments.

In this scenario, the C&DH system is the single IP address to which, and from which, all data (commands and telemetry) are routed. The C&DH system acts as both an initial command interpreter and software router to transfer the commands to the flight software tasks and the instruments. Similarly, the C&DH acts as the sender of all spacecraft engineering data and instrument payload data back to the ground station. The only change to the heritage system is the modification of the C&DH system to include the IP component.

Figure 3–1 provides a representative example of this type of on–board architecture.

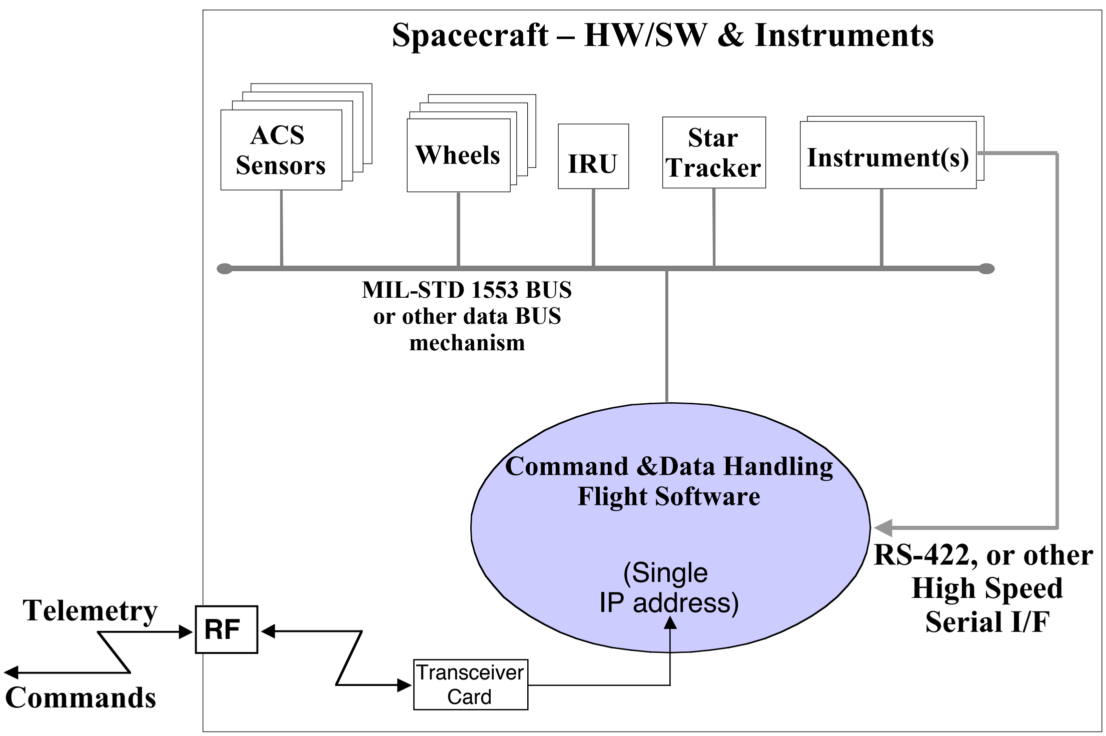

Figure 3–1 Legacy spacecraft Architecture using Single IP Addressing

3.2.1.2 A Slightly More Complex Solution

This approach is slightly more complex than the previous example, but does not introduce significantly more complex features. For this approach, we could assume a wide variety of spacecraft orbits, from the LEO spacecraft example previously noted to even more advanced medium earth orbiting (MEO), to high earth orbiting (HEO) spacecraft.

A more complex interaction would result from a requirement for TDRSS–supplemented ground station contacts, which require interaction with the White Sands Complex at various times in addition to the standard ground station contacts. With this approach, the SE still specifies standard IP connectivity with predefined wide and local area networks. The SE would identify trade studies to determine the requirements for data transfer, including reliability requirements, and determine which communications are routed as UDP/IP messages and which as TCP/IP messages. Section 4 provides more details on these trade study concepts.

In this scenario, the C&DH system is the single IP address to which, and from which, all data (commands and telemetry) are routed. The C&DH system acts as both an initial command interpreter and software router to transfer the commands to the flight software tasks and the instruments. Similarly, the C&DH acts as the sender of all spacecraft engineering data and instrument payload data back to the ground station.

This approach now uses some type of on–board LAN to communicate with the other spacecraft subsystems or instruments. The on–board LAN could be configured with a different set of IP addresses for each subsystem or instrument, all which are on a single subnet.

Figure 3–2 provides a representative example of this type of on–board architecture.

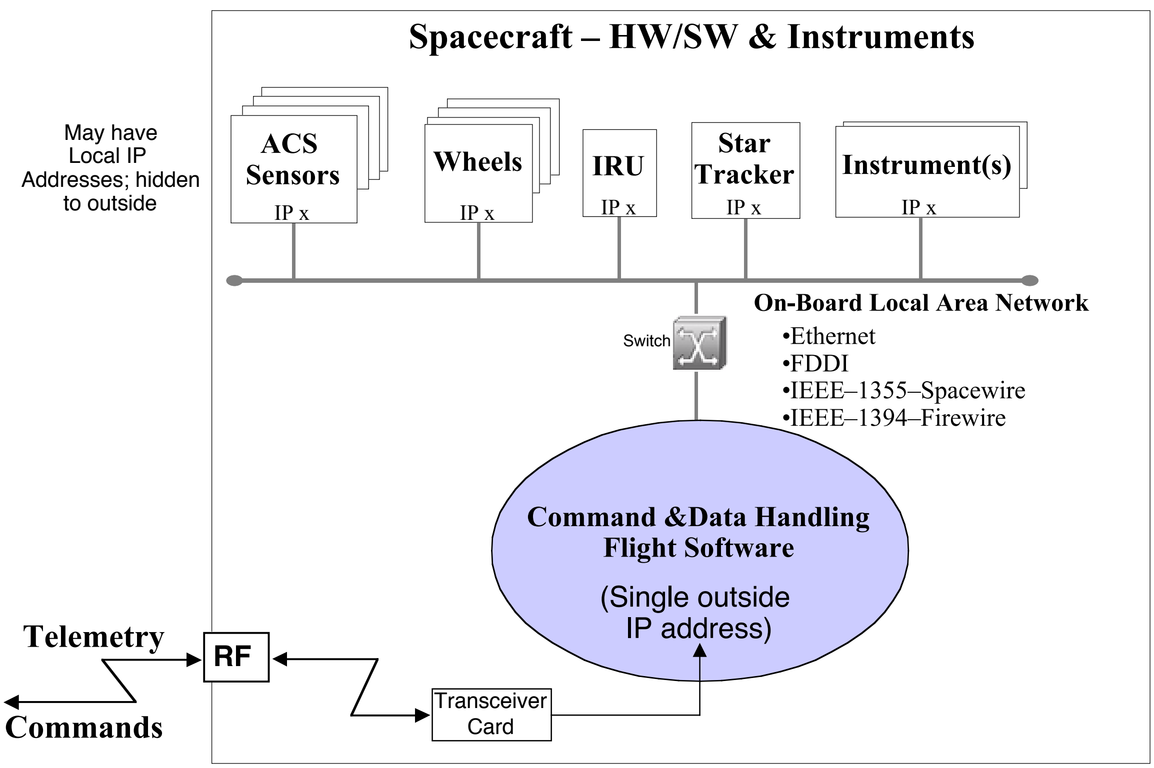

Figure 3–2 Spacecraft Architecture using Single IP Addressing

3.2.1.3 Multiple IP Addressing Architecture

The final instantiation is a complex approach that defines the requirement for Mobile IP/mobile routing, and includes multiple IP addresses on–board the spacecraft, supplemented by an on–board Ethernet LAN and space–hardened routers.

In this scenario, the entire spacecraft has a set of independent IP addresses, to which command and telemetry data can be independently routed. The number of unique IP addresses is only limited by on–board IP stack and data transfer rates.

This approach continues the use of an on–board LAN to communicate with the other spacecraft subsystems or instruments. The on–board LAN would be configured with the set of IP addresses for each subsystem or instrument.

The SE would identify trade studies for data reliability by documenting the requirements needed, if any, for redundant on–board HW systems, particularly the LAN and routers/switches. The SE also would provide the operations concepts needed to ensure that all command data is reliably transmitted to the spacecraft and would identify the data throughput and completeness requirements to ensure that spacecraft data is delivered to the mission and science operation centers.

Figure 3–3 provides a representative example of this type of on–board architecture.

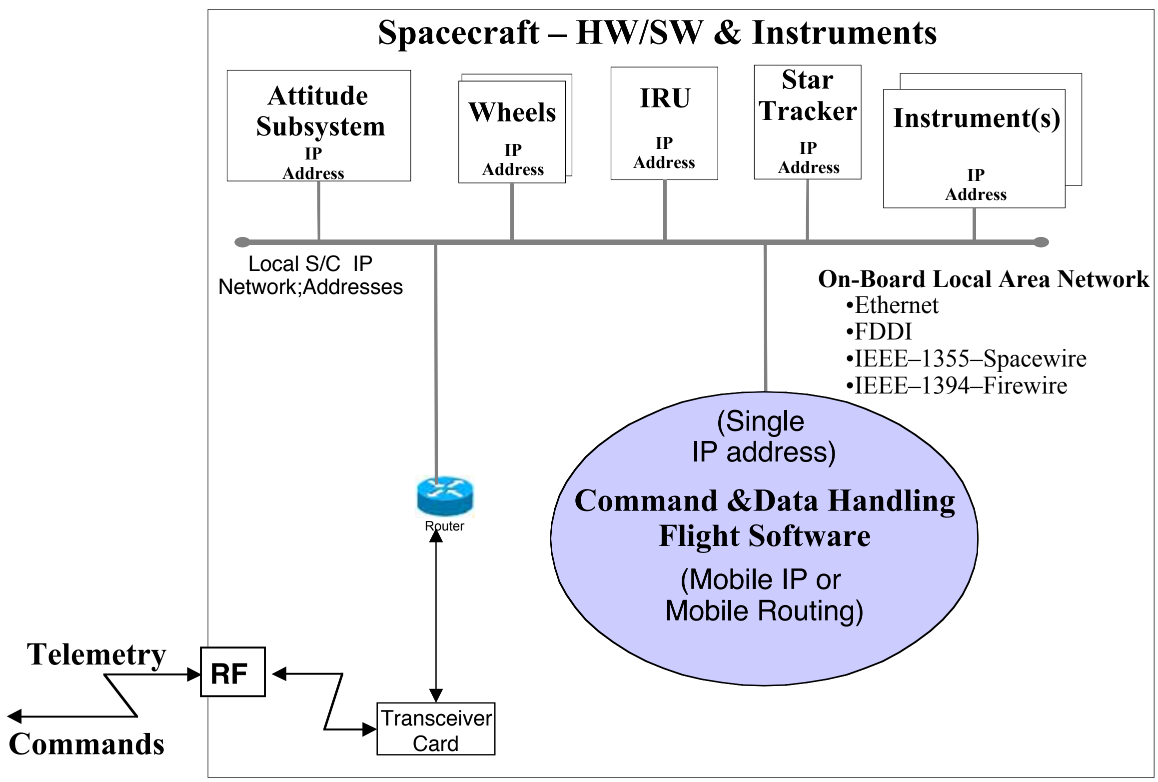

Figure 3–3 Spacecraft Architecture using Multiple IP Addressing

3.2.1.4 More Complex Mission Scenarios

The following aspects will be refined and updated as additional details become available. This sub–section will expand to meet the needs and requirements of future missions and the System Engineers assigned to support the missions.

- Mobile IP and Mobile routing with multiple ground stations or with the TDRSS constellation.

- High earth orbit or deep space mission (round–trip delay times), space–to–space cross–links (what's considered uplink versus downlink; many–to–many, peer–to–peer, many–to–one, or one–to–many communications).

3.2.2 Space–Segment, Flight Qualified IP Components

To fully support a complete end–to–end IP implementation, the space segment system must include radiation–hardened components that support the implementation of the LAN (which supports the communications between the spacecraft subsystems) and the WAN for the RF/Optical link interface (which supports the communications between the spacecraft and the ground station(s)). This conceptual architecture is represented in Figure 3–4 and provides a brief view into the IP hardware components that are necessary for a complete end–to–end implementation approach.

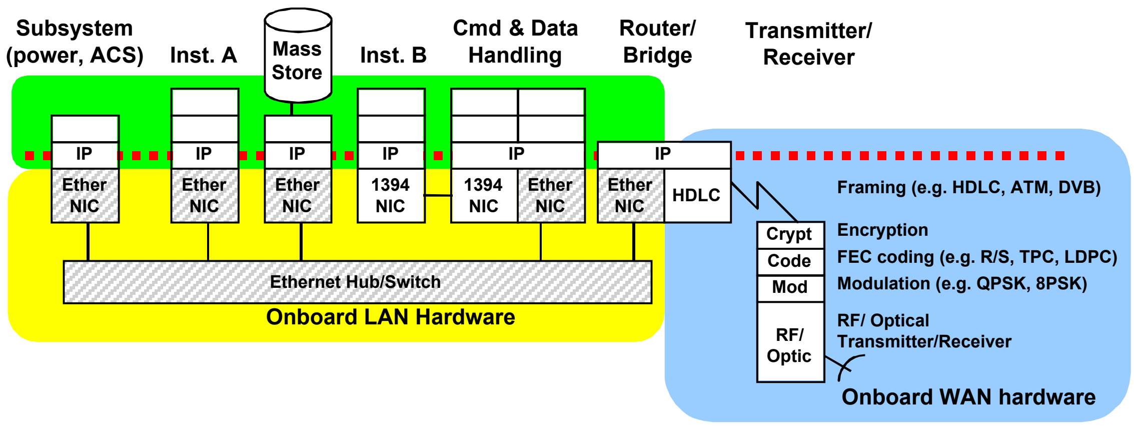

Figure 3–4 Conceptual Space Segment HW Architecture Components

3.2.2.1 Space Segment Local Area Network Components

Currently (as of September 2003), various groups are supporting the development of several hardware components required for the space segment. These components must be flight–qualified, or hardened against the strict radiation environment of space, or must ensure the fault tolerance of the components such that no one single failure could result in mission failure. The following table provides a brief review of the current HW components and their associated status:

At the time of this writing (September 2003), several LAN card developments are currently under way and should provide 100 Megabits per second Direct Memory Access (DMA) capable connectivity; these include radiation–hardened Ethernet network interface cards (NICs) developed by Spectrum Astro, IEEE–1394 Firewire NICs, and ATM NICs. The Jet Propulsion Laboratory (JPL) X2000 IEEE–1394 interface board – Flight Application–Specific Integrated Circuits (ASICS) also is currently in work. The JPL interface board will likely be the first flight–qualified card but probably will be used by high–end customers.

GSFC and European Space Agency (ESA) are working on an IEEE–1355/Spacewire concept. This flight qualified NIC card will likely be possible by 2004. The GSFC/ESA current activity is developing a breadboard Ethernet LAN card. In principle, IP even could be transported across standard serial or even MIL–STD–1553 interfaces.

Table 3–2 Space–Segment LAN Hardware Component Status

Hardware Component	Current Status (As of September 2003)
Space Qualified Ethernet, Firewire, ATM Connectors	Rugged Ethernet connectors for factory floor
Radiation–hardened Ethernet 10/100/1000 Network Interface Cards (NICs)	Spectrum Astro working on 10/100 Ethernet using rad–hardened, qualified COTS components NASA/GSFC building rad–hardened FPGA Ethernet
Rad–Hardened Firewire (IEEE 1394) NICs	Ball Aerospace developing for NPOESS mission
Rad–Hardened ATM NICs	Northrup–Gruman (TRW) Astrolink design
Rad–Hardened Ethernet, ATM Hubs, Switches	NASA/GSFC working on the Ethernet Switch
Device Drivers for NICS in standard OS	Should be similar to standard NICs
Fault tolerant LAN equipment and failure recovery strategies	Factory automation and process control community working to complete components and concepts
High–speed, network attached random access mass storage for file systems	Possible application of Storage Area Networks (SAN) and iSCSI network storage concepts
High stability, radiation–hardened, time systems (clocks, network time servers)	Work needed on low–cost, stable clocks

The Glenn research center is working with various industry groups on a space router that supports IPsec, Mobile IP, and Mobile Routing. GSFC & ITT Industries have developed a router for the Low–power Transceiver (LPT) digital radio Shuttle (STS107) flight for the CANDOS mission.

3.2.2.2 Space Segment Wide Area Network Components

Currently (as of September 2003), various groups are supporting the development of several hardware components required for the space segment. These components must be flight–qualified, or hardened against the strict radiation environment of space, or must ensure the fault tolerance of the components such that no one single failure could result in mission failure. The following table provides a brief review of the current HW components and their associated status:

Table 3–3 Space–Segment WAN Hardware Component Status

HW Component	Status
Link–encryption/decryption HW	Some military components available
Radiation–hardened, forward error correction HW (e.g., Reed/Solomon, Low–density parity check, Turbo Product Code	R/S encoders available for space Work underway for both LDPC and TPC
Radiation–hardened framing HW (e.g., HDLC< ATM, EIA IS–787, DVB	COTS HDLC chips used on LEO spacecraft over last 20 years; simple to implement in rad–hardened FPGAs
High–rate versions of coding, encryption, and framing HW	Possible solutions under development from DoD Transformational Communications project
Rad–hardened Ethernet, ATM bridges to transmitter/receiver	NASA/GSFC mission (Global Precipitation Measurement) developing Ethernet/serial bridge
Rad–hardened routers with Ethernet, Firewire, ATM serial interfaces	General Dynamics (Motorola) and Cisco developing prototype ITT added routing features to LPT
Basic mobile routing protocols	Mobile IP available and flown on STS–107 (CANDOS) More mobility solutions will be available in IPv6
Mobile routing to hide mobility details from on–board systems	Cisco Mobile routing being developed for test flight on SSTL Disaster Monitoring Constellation spacecraft in 2003

3.3 Ground Segment Architecture Analysis

This section describes how the ground elements are developed using an IP implementation approach. It discuses the ground station and how it interacts with the MOC and SOC as well as the spacecraft It briefly discusses the mission operations center (MOC), which houses the command and control components required to operate the spacecraft as well as any ancillary elements required to support the mission. It discusses the science operations center (SOC), which is responsible for the daily science planning and coordination as well as the daily processing of the received raw science telemetry data.

Similar to Section 3.2.2, the ground segment will also have WAN/LAN technologies that support the complete end–to–end IP implementation; Figure 3–5 provides a conceptual view of the hardware components that support the ground receipt and transfer of data. The main objectives in the ground segment are to identify solutions (modulation, FEC coding, encryption, and framing mechanisms) that are widely available, efficient, and feasible for any/all remote systems.

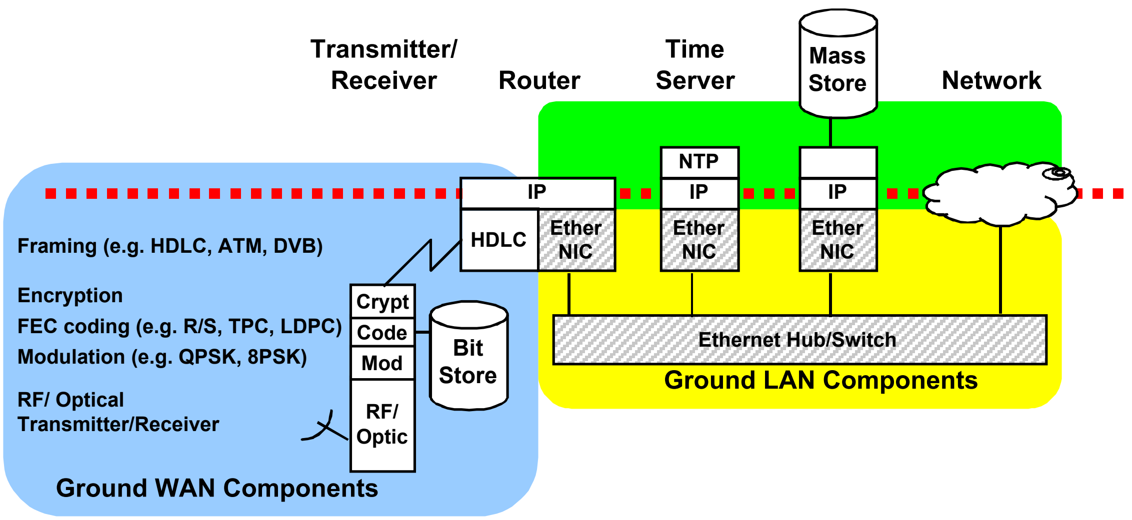

Figure 3–5. Ground Segment Conceptual HW Architecture

The following table provides an overview of the status for those HW components associated with the ground segment.

Table 3–4. Ground System HW Component Status

HW Component	Status (As of September 2003)
High–rate forward error correction HW (e.g., Reed/Solomon, Low–density parity Check, Turbo Product Code	R/S and Turbo code available LDPC becoming popular Gbps rate support still needed for all FEC HW
High–rate framing compatible with remote system framing (e.g., HDLC, DVB, ATM, etc.)	HDLC framing available up to 100 Mbps DVB framing available up to 240 Mbps Requires space HW at any higher rate
High–mass storage system	Large storage area networks (SAN) systems available with increasing capacity and speed
Network Timing systems for use by s/c	Standard NTP time servers are available but may need special features for higher precision timing and to accommodate RF/optical links
Standard high–rate–ground networking technologies	Available at rates of 10 Gbps and is constantly growing

HW Component	Status (As of September 2003)
Mobile routing protocol	Mobile IP protocol flown by CANDOS mission (STS–107, Jan 2003) Mobile routing technologies being deployed (to be flown late in 2003 IPv6 evolving in test environments (e.g., 6Bone – IPv6 Testbed)
End–to–End IP addressing concepts	Work still needed on scalable system addressing schemes and operational security mechanisms
Security Approaches	Commercial security options available, but Scalable addressing and security needs to be developed for range and space applications.

3.3.1 Ground Station – Conceptual Architecture and IP Revisions

The ground station is the central hub in the communications between the ground and space segments. This section provides a conceptual representation of the ground station and the interfaces with both the ground and space segments. Figure 3-6 represents a conceptual view of this architecture. In this example, any interface from a ground component, normally the control center will end at the *"Ground IP Routing"* component. The ground station will remove any "ground artifacts" and ready the data for transmission to the spacecraft, using the *"Space IP Routing"* component; this component is discussed in more detail in the following paragraphs of this section.

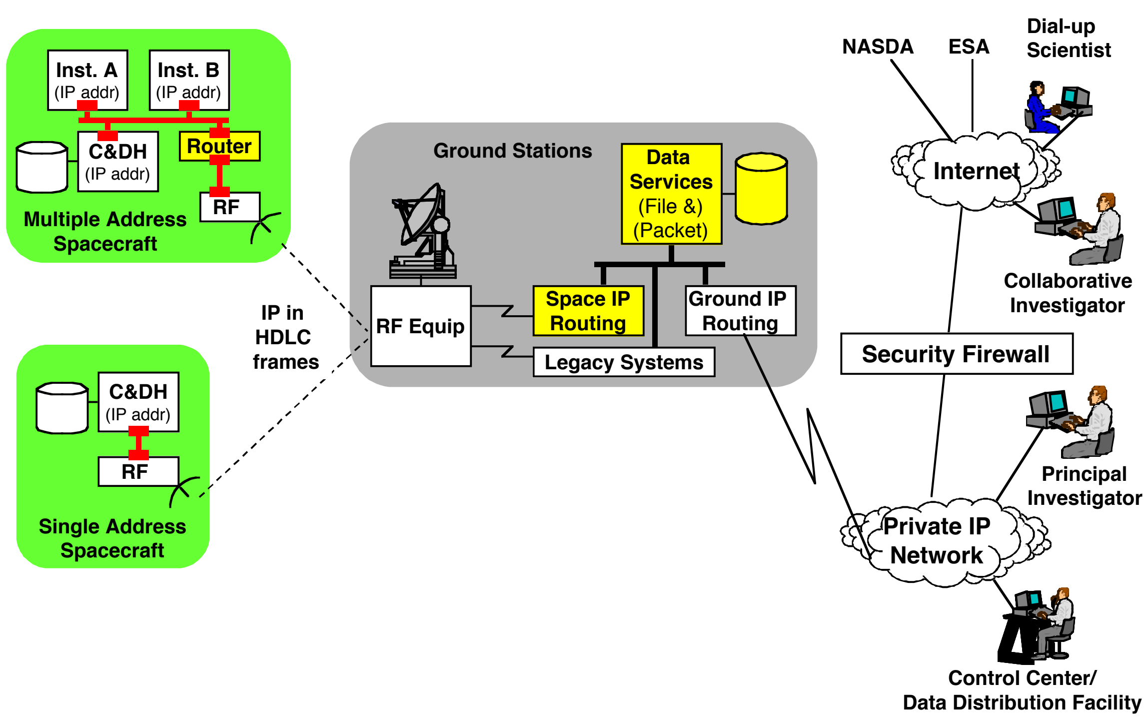

Figure 3–6 Example of Ground–Space Internet Physical Components

Legacy equipment in the ground stations will not interface directly to a commercial router to support an IP space link. A device similar to a commercial satellite modem is needed to connect the RF interfaces to commercial routers. This approach is depicted in Figure 3–7, where the device is identified as the Ground Station Router Interface Device or GRID.

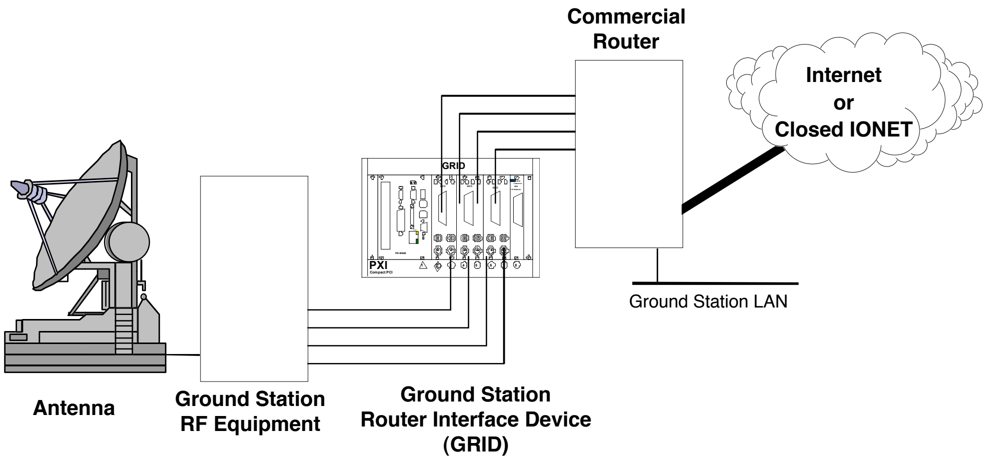

Figure 3–7 Ground Station Modifications

The SE would identify what HW and SW modifications are required at the ground station to support the mission. For example, the SE would indicate whether all data is transmitted in real–time. If so, then no file transfer concepts are required to be included at the ground station. Alternatively, if the SE determines that file transfers will be employed, then the station may be required to support the file transfer concept and/or a store–and–forward concept.

The SE may require that the ground station implement a multicast capability to transmit the data to multiple addresses, such as the ground control center and various science centers. The SE would identify the requirements associated with data completeness if there are file transfers and several ground stations supporting the mission. The SE would determine how files are transferred from the spacecraft to each station and identify the requirements needed to support file transfer associated with station handovers and how that concept is used to support the data completeness science requirement.

The SE would perform the trade studies to identify whether mobile IP/routing is needed to support the overall mission data connectivity requirements. If the mission has a dedicated ground station, or is in continual view of one station, then no mobile IP requirements are needed. If the spacecraft needs several stations to support the data connectivity/completeness requirements, then the SE would document the requirements for mobile IP/routing and would document the requirements for station handovers and the means and operational procedures for transferring data (either R/T or file from one station to another.

3.3.2 Mission Operations Center (MOC)

The MOC is the central facility responsible for the daily spacecraft operations; it receives the s/c telemetry and performs various trending and analysis functions to identify any short- or long-term aspects of the HW systems. The MOC will distribute the ancillary data needed by SOCs.

The SE would determine whether all instrument commands must go to the MOC to preclude violations between multiple command loads for different instrument groups; the SE would determine whether all data (spacecraft and instrument data) must be transmitted to the MOC; if some version of a file transfer protocol is used, the SE determines the location of the terminus component on the ground (at the MOC, ground station, or at each individual SOC).

3.3.3 Science Operations Centers (SOCs)

There may be one or more SOCs located at various facilities, connected via a LAN/WAN. The SE would determine whether to allow the SOCs to directly command instrument(s); who has the responsibility for ensuring that instrument commands for one instrument do not cause problems for another instrument; how the distribution of science data to other users is accomplished; and what the SOC requirements for the receipt of spacecraft ancillary data such as attitude and orbit data are, and it is generated from the MOC or transmitted directly from the spacecraft.

3.4 Ground to Ground Data Transfer and Conceptual Components

Currently, NASA ground–to–ground interfaces are provided via an instantiation of an IP network. The security branch completed an analysis of the standard interface systems used within the NASA environment; they classified the following sets of networks that can be used for any mission. This analysis is documented in their IP-in-Space Security Handbook; Executive Summary; Section 1.4.2 *"Networks"*. As a refresher, that information is:

Closed Mission Network – that network, which has been certified by NASA as meeting the security requirements as defined in NPG 2810.1 for mission (MSN) information; it is connected only to an "Open Mission Network" by a stateful firewall.

Open Mission Network – that network, which has been certified by NASA as meting the security requirements as defined by NPG 2810.1 for MSN information; but it is connected to other networks that may not fulfill the NPG 2810.1 security requirements for MSN information.

Open Network – that network, which has not been certified by NASA as meeting the security requirements for MSN information, and is accessible by persons unaffiliated with mission operations.

The command or telemetry data are transferred using either a TCP–based or a UDP–based application. One tradeoff that the SE is required to perform is whether all components are on an "open" Internet or whether some components, such as the control center, are hidden behind a security firewall on a "secure" network. Figure 3–8 provides an example in which several components are located behind their own respective security firewalls. In this example, all ground–initiated communications with the spacecraft terminate at the "Ground IP Routing" component located at the ground station; this concept was discussed in Section.3.3.1. Additional details regarding the security requirements and possible trades are discussed in more detain in Section 5

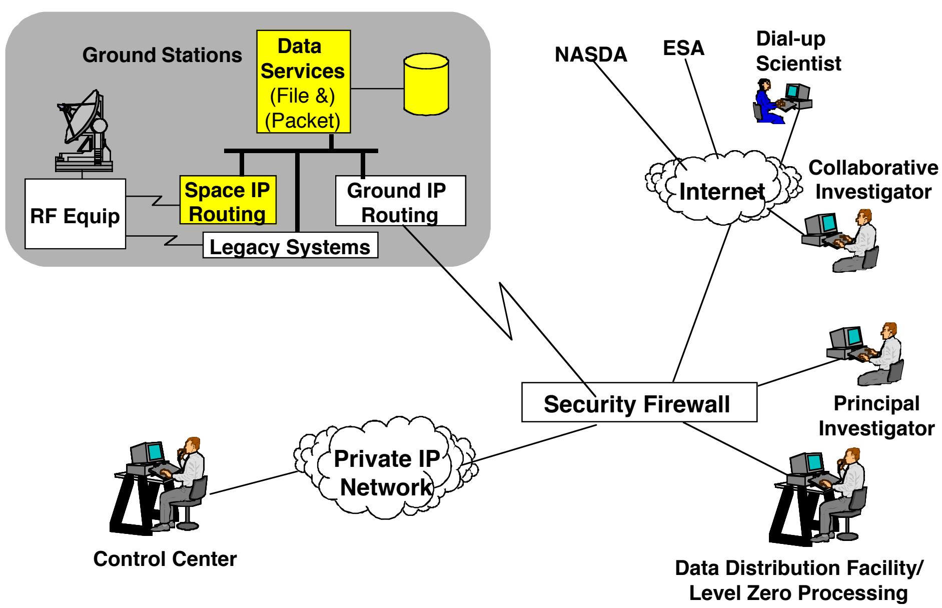

Figure 3–8 Example of Ground Internet Physical Components

Another trade study that the SE would perform is to determine the quality of service (QoS) required to support the data rate and throughput requirements. The Internet, as originally conceived, offered only a very simple QoS point–to–point, best effort for data delivery. Within this service profile the network would make no attempt to actively differentiate its service response between the traffic streams generated by concurrent users of the network. As the load generated by the active traffic flows within the network varies, the network's best effort service response also will vary.

3.4.1 Quality of Service Introduction

Before real–time applications such as remote video, multimedia conferencing, visualization, and virtual reality could be broadly used, the Internet infrastructure had to be modified to support real–time QoS, which provides some control over end–to–end packet delays. This extension had to be designed from the beginning for multicasting; simply generalizing from the unicast (point–to–point) case does not work.

QoS encompasses several technologies that enable selected applications to receive preferential access to shared resources, thus assuring a specified level of performance for these applications. Pertinent performance parameters include jitter, throughput, latency, and packet loss. QoS is the ability to provide consistent, predictable data service delivery to satisfy customer requirements. Several characteristics qualify QoS, including the capability to minimize delivery delay, reduce delay variations, and provide consistent data throughput capacity.

There are various approaches to allocating resources to ensure that values of these performance parameters stay within acceptable ranges for specific applications or classes of applications. The following two examples are ways in which the QoS can be implemented:

- Priority Queuing
- Network Bandwidth Allocation and Reservation

Each of the above listed terms is presented in an introductory fashion here. A more detailed description is provided in Appendix A.9; that section provides more information on the terms and their uses and discusses relevant issues and techniques for ensuring traffic flow without any degradation in data throughput.

3.4.2 Priority Queuing

Priority queuing supports a small number of queues, usually from high to low priority. Queues are serviced in strict order of priority, so the high queue always is serviced first, then the next–lower priority, and so on.

If a lower–priority queue is being serviced and a packet enters a higher queue, that higher queue is serviced immediately. This mechanism is good for important traffic, but can lead to queue starvation.

3.4.3 Network Bandwidth Allocation and Reservation

This refers to dedicating bandwidth to a specific application or set of applications. The customer, using these application(s), pays for the dedicated bandwidth regardless of how often, or how little, the bandwidth is actually used. Because bandwidth is dedicated, other applications cannot use it, even when there is little congestion on the connection. This usually is done when other QoS methods, such as weighted fair queuing, are not effective in providing the desired handling of priority traffic. This approach is used more in the closed network domains, such as those operated by NASA.

3.5 Space to Space Data Transfers

The main trade study analysis that the SE will conduct relates to the ad–hoc networking concepts involved with communications between orbiting spacecraft -- whether the spacecraft are flying in a close formation, constellation flying over a wide displacement in varying orbits, or acting as a relay to a dedicated ground station (e.g., TDRS to White Sands) or even to another orbiting spacecraft.

SECTION 4 OPERATIONAL SCENARIOS

This section defines operational scenarios for missions, and offers guidelines for IP–based implementations. It also provides additional details on orbital concepts and TCP/IP that the SE will need in order to define any derived requirements for on–board memory.

4.1 Space – Ground Data Transfers (Uni– and Bi–Directional)

4.1.1 Real Time Transfers

Real–time data can consist of either commanding or telemetry. Each is discussed in detail in the following sections.

4.1.1.1 Real Time Commanding

Real time commanding typically operates in one of two modes: confirmed or blind. In confirmed commanding (sometimes known as two–step commanding), the receipt of each command, in sequence, is confirmed before the next command is accepted and executed. Additionally, the result of each command, if any, may be confirmed via telemetry before the next command is sent. In blind commanding, the command is sent over a forward link, without the use of any return link for confirmation.

4.1.1.1.1 Confirmed Commanding

For near–earth and sub–orbital missions, the SE has the option of choosing either UDP or TCP for confirmed real–time commanding. For missions with highly elliptical orbits (apogee greater than ~12 earth radii), or deep space missions, previous analysis of the bandwidth delay product has shown that UDP is indicated due to the large transmission delays.

Both UDP and TCP each support up to 65535 unique ports. Each commandable system or subsystem should be assigned a unique port number. This allows the built–in IP stack to accomplish all of the multiplexing and de–multiplexing of interleaved commands to multiple systems. Port numbers under 1024 are typically reserved for the operating system, and many port numbers under 8000 are reserved and pre–defined for specific services.

If UDP–based confirmed commanding is chosen, the SE would typically specify an application–level handshake be implemented on top of UDP. Any additional command confirmation via telemetry values would have to be accomplished either manually or by application–level software; this is functionally identical to non–IP based missions.

If TCP–based confirmed commanding is chosen, the built–in TCP/IP stack supplies most of the needed functionality via CRC's and retransmissions. This ensures that the spacecraft reliably receives the correct command bytes, in the correct order. If the SE requires any additional TCP–based command confirmation via telemetry values, the SE would have to levy requirements against some application–level software or would have to mandate that this confirmation is accomplished manually by mission operations personnel. Command uplink rates are normally 1–2 Kbps (or less) for most sub–orbital and earth–orbiting spacecraft. At near–earth distances, these command rates are easily supported by the standard TCP windows. See Appendix E for a discussion of TCP window size in order to determine memory buffer requirements.

When the delay and buffering requirements are too great for the mission to support with either a TCP–based or UDP–based confirmed commanding approach, the SE should choose to incorporate either a blind commanding or stored commanding approach to meet the mission requirements.

4.1.1.1.2 Blind Commanding

All spacecraft must have some minimal amount of blind commanding, usually implemented in hardware, to allow for recovery from spacecraft anomalies. This usually consists of a 'reset' command but may include other fundamental operations such as 'safing' the spacecraft. In addition, any command could optionally be required to be sent as a blind command. This can occur if the mission profile includes significant amounts of forward–link–only contacts, or if the commands are stand–alone and are not constrained by (and do not constrain) any other command, or if the commands are required to configure the downlink. In these cases, the ground control center does not wait for a command acknowledgement/confirmation.

The SE would typically choose UDP to support blind commanding. Hardware blind commands can be embedded inside of a UDP packet as well. The hardware decoder ignores the surrounding UDP/IP headers and simply examines the incoming byte stream for the unique hardware commands. This allows for supporting hardware commands while still retaining the automatic routing of capability IP packets. Separate UDP port numbers can be used to further segregate blind commands if required.

4.1.1.2 Real Time Telemetry

Real–time telemetry can consist of any combination of spacecraft–bus engineering data, payload–engineering data, and live science data.

4.1.1.2.1 Unidirectional – UDP

For most IP missions, real–time telemetry will be carried in UDP packets, and the SE would document the requirement for UDP telemetry. UDP, also sometimes called unreliable delivery, is no different from the current approach that is used today. Packets are "send and forget". There is no guarantee of the data delivery; you simply receive what you receive in the space–to–ground link without any automatic retries on the part of the sending spacecraft. This is a common aspect of real–time telemetry, where the most current values are more important than any past missed values.

UDP, documented in RFC 768, provides users with access to IP services. UDP packets are delivered inside of IP packets. The UDP packets are connectionless datagrams. Like IP, the UDP datagrams may be discarded before reaching their targets. UDP is useful when TCP would be too complex, too slow, or just unnecessary. UDP *does* checksum its data, ensuring data integrity. A packet that fails the checksum is simply discarded, with no further action taken. This UDP–based checksum protection is *in addition to* the protection provided by the IP header checksum and the link–layer frame CRC. Because of this, UDP packets are "atomic' in nature; that is, either the entire packet is delivered intact, or none of the packet is delivered. There are no partial packets with UDP.

Past missions have segregated the real–time telemetry into "virtual channels", with possible further segregation within a virtual channel by "application ID". UDP has a maximum of 65535 unique "port" numbers available. This mechanism supports the capability for the SE to assign a unique port number to each possible set of spacecraft and science telemetry mnemonics. This could eventually be used to support the independent routing of data to various mission and science operations centers.

4.1.1.2.2 Reliable – TCP

The SE would document the requirement for TCP–based reliable downlink of telemetry only when the following conditions exist:

- A requirement exists for lossless error–free real–time telemetry
- A 2–way link exists
- The round–trip delay for retransmission is within the acceptable TCP maximum

This is an unusual configuration, and any apparent requirement for it should be examined with great care. The real–time nature of telemetry usually implies that having the most current data values is more important than having every data value. Frequently, the apparent requirement for "lossless error–free real–time telemetry" will actually be found to decompose into two separate requirements: "Real–time Telemetry" (as defined in Section 4.1.1.2) and "Downlinked Stored Telemetry and Science Data" (as defined in Section 4.1.2.2).

Given that the requirement is found to be genuine, the SE would calculate the bandwidth–delay product, as defined in Appendix E, in order to specify the required TCP window size and generate the derived requirement for the necessary onboard memory. The standard TCP window size allows for buffer sizes up to 64 Kbytes. The widely available "TCP Large Windows" option (RFC–1323) increases this maximum up to ~1 Gbyte. Separate ports can also be used to spread the buffer requirements out over multiple TCP connections if "TCP Large Windows" is not available.

In cases where the round–trip delay times are unacceptable for TCP, the SE should choose a UDP–based reliable streaming protocol. Examples of these are the Multicast Dissemination Protocol (MDP), NACK Oriented Reliable Multicast (NORM) or the CCSDS File Delivery Protocol (CFDP).

4.1.2 Stored Data Transfers

Stored data transfers are based on files. In most cases, this allows the flight operating system's native file system to handle the low–level storage management and allocation with standard, well–characterized code. In those situations where throughput or latency requirements exist that exceed the capabilities of the built–in file system, it may be necessary to specify a custom file system (with a standard interface) that meets those requirements.

4.1.2.1 Uplink Command Loads and Tables

Commands consist of any information sent to the spacecraft that are either acted upon in real–time or are stored for future execution. These commands can consist of any of the following types of data:

- Immediate Spacecraft bus commands;
- Immediate Payload/Observatory commands;
- Time–Tagged Spacecraft bus commands;
- Time–Tagged Science commands;
- Software uploads or patches
- Table Loads / Table updates
- All of the above as email attachments

4.1.2.1.1 Short Delay

In most low earth orbit missions, the SE would choose a TCP–based protocol, such as the File Transfer Protocol (FTP) or the Secure Copy Protocol (SCP), for the uplink of command loads and tables, unless either of the following conditions apply:

- Residual bit–error–rate (BER) is worse than 10^{-5} <u>after</u> any physical layer forward error correction (FEC)
- Bandwidth allocation requirements restrict the maximum instantaneous percentage of the uplink bandwidth that command load uplinks can consume

If either of these conditions exists, the SE would chose a UDP–based reliable file transfer protocol that uses negative acknowledgements and can be throttled, such as MDP, NORM, or CFDP.

The short delay reference identifies that there is only a small transmission delay period between the spacecraft and the ground station. Because of the typically low (2 – 4 Kbps) uplink rates and the short transmission delay, the TCP window size and link asymmetry are generally not issues. The effective throughput (aka 'goodput') of TCP–based file transfer protocols is sensitive to the residual bit–error–rate (BER) <u>after</u> any physical layer forward error correction (FEC) has been applied. At a residual BER of 10^{-7}, goodput remains close to 100% of available bandwidth. At a residual BER of 10^{-}

^{5}, goodput can drop to as low as 33% of available bandwidth. For residual BERs worse than 10^{-5}, TCP–based file transfer protocols are not recommended.

4.1.2.1.2 Long Delay

The SE would select a UDP–based reliable file transfer protocol for the uplink of command loads and tables when the mission design incurs large transmission delays. This would occur for missions with highly elliptical orbits (apogee greater than ~12 earth radii), or deep space missions. Possibilities include MDP, NORM, and CFDP (Refer to Appendix D).

4.1.2.1.3 Store & Forward – SMTP over TCP or BSMTP over MDP/UDP

Simple Mail Transfer Protocol (SMTP) is a protocol for delivering email, plus attachments, from one system to another. Originally designed for delivery across intermittent dial–up connections, it lends itself to space mission systems requiring a store & forward capability for the uplink of command loads and table updates.

When TCP–based file transfers are possible, as discussed in Section 4.1.2.1.1–Short Delay, SMTP can be used as is. When UDP–based file transfers are required, batch SMTP (BSMTP) can be used to batch up the messages interleaved with SMTP commands, and a UDP–based reliable file transfer used to deliver the file.

4.1.2.2 Downlink Stored Telemetry and Science Data

This section defines the requirements to support a reliable delivery of stored spacecraft bus and engineering data and stored payload engineering and science data. This option is used whenever the SE has defined that not all data is delivered in real–time because of station contact limitations or bandwidth limitations. In this event, some amount of data would be stored on board for a later downlink to a ground station.

4.1.2.2.1 Short Delay – FTP or SCP over TCP

In most low earth orbit missions, the SE would choose a TCP–based protocol, such as the File Transfer Protocol (FTP) or the Secure Copy Protocol (SCP), for the downlink of stored data files, unless any of the following conditions apply:

- Residual bit–error–rate (BER) worse than 10^{-5} <u>after</u> any physical layer forward error correction (FEC)
- Bandwidth allocation requirements restrict the maximum instantaneous percentage of the downlink bandwidth that data file downlinks can consume
- The required downlink bandwidth is greater than 50 times the available uplink bandwidth

If any of these conditions exist, the SE would chose a UDP–based reliable file transfer protocol that uses negative acknowledgements (NACKs) and can be throttled, such as MDP, NORM, or CFDP, as discussed in Section 4.1.2.2.2–Long Delay.

The effective throughput (aka ‘goodput’) of TCP–based file transfer protocols is sensitive to the residual bit–error–rate (BER) <u>after</u> any physical layer forward error correction (FEC) has been applied. At a residual BER of 10^{-7}, goodput remains close to 100% of available bandwidth. At a residual BER of 10^{-5}, goodput can drop to as low as 33% of available bandwidth. For residual BERs worse than 10^{-5}, TCP–based file transfer protocols are not recommended. TCP ‘goodput’ begins to suffer when the link asymmetry exceeds 50:1. NACK–based protocols operate efficiently with link asymmetries in excess of 1000:1. Appendix E–1 provides more information on TCP link asymmetry requirements.

4.1.2.2.2 Long Delay

The SE would select a UDP–based reliable file transfer protocol for the downlink of stored data files when the mission design incurs large transmission delays. This would occur for missions with highly elliptical orbits (apogee greater than ~12 earth radii), or deep space missions. Possibilities include MDP, NORM, and CFDP.

4.1.2.2.3 Store & Forward – SMTP, MDP or CFTP

This section is identical with the store and forward concepts defined for commanding, which was described in Section 4.1.2.1.3–Store & Forward – SMTP over TCP or BSMTP over MDP/UDP.

4.1.3 Onboard Clock Synchronization

Onboard clock synchronization is required in order to keep all spacecraft and payload subsystems locked on an exact time so that instrument data can be correlated to other spacecraft ancillary data such as attitude. Two IP techniques can be used to measure and correct the onboard clock bias. They are Network Time Protocol (NTP) and time–stamped ICMP packets.

4.1.3.1 Synchronization and clock drift mitigation – NTP

The Network Time Protocol (NTP) is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver or modem. It can provide accuracies to within microseconds dependant upon adequate link bandwidth and OS CPU speed. The time accuracy is relative to Coordinated Universal Time (UTC) via a Global Positioning System (GPS) receiver, for example. Typical NTP configurations utilize multiple redundant servers and diverse network paths in order to achieve high accuracy and reliability. Some configurations include cryptographic authentication to prevent accidental or malicious protocol attacks, and some provide automatic server discovery using IP multicast.

4.1.3.2 Time–stamped ICMP Packets

The Internet Control Message Protocol (ICMP), defined in RFC 792, is a part of every standards–compliant IP stack. It provides for an echo–request/echo–response set of packets that can be time–stamped at each end of the transaction. The time–stamps record each system clock's time in milliseconds past midnight, to a resolution of 1 msec. When used between the spacecraft and a system located at the ground station, the speed–of–light propagation delays are known and can be factored out, yielding the offset of the spacecraft's clock from the ground station system. This offset can be used to manually reset the spacecraft's clock, or can be the input to an automated application, running either on the ground or on the spacecraft.

4.1.4 Spacecraft Orbit Concepts

Some of the derived requirements for a mission are driven by the mission's orbital parameters. Round–trip delay time, in particular, constrains the choice of TCP vs. UDP transports, and determines the amount of TCP buffer memory required.

Reliable data delivery means that, for every packet (or group of packets) of data transferred by the spacecraft, the ground station acknowledges that it received the packet(s). This limits the volume of data that the spacecraft can downlink before it must wait for the ground acknowledgement. When using TCP for reliable data delivery, this buffering is provided by the operating system's TCP/IP stack, and is called the TCP–window size. When using a UDP–based method for achieving reliable delivery, the application program must provide this buffering.

Table 4–1 Approximate Spacecraft Orbital Data

Sample Spacecraft orbit	Approximate Distances		Approximate Light Speed Roundtrip Delay (seconds)
	Km	Miles	
Sub–orbital	120	75	0.00081
LEO	600	375	0.00403
MEO	6000	4125	0.04435
5 R_e	32000	20000	0.10753
GEO	36000	22500	0.24194
60 R_e	382685	239178	1.28590
Sun–earth Lagrange 1 point	1500000	937500	10.80645
Mars at conjunction	78400000	49000000	526.88172
Mars at opposition	376000000	235000000	2526.88172

4.1.4.1 TDRSS Relay

Spacecraft that are in Low Earth Orbit (LEO) or Medium Earth Orbit (MEO) have the option of sending their communication through TDRSS. Because the TDRSS satellites are at geosynchronous altitude, the approximate round–trip–delay time is 0.25 seconds. For a given data rate, this increases the bandwidth–delay product, and thus the TCP window buffer size, as described in Appendix E. It also provides for longer contact times, and thus larger total data volumes, than a direct ground station.

4.1.4.2 Direct Ground Station

All missions, either near–earth or deep space, can elect to utilize a direct ground station. Near–earth missions will experience much shorter round–trip delay times when using a direct ground station instead of TDRSS, but will be constrained to contact times on the order of 10 minutes. This limits the total volume of data transfer possible.

Deep–space missions could choose to use either the Deep Space Network (DSN) or partner with the European Space Agency and use their 34–meter stations. This approach is required due to the large distances and low received signal strength. The low signal strength tends to limit the maximum data rate to lower values than near–earth missions. Although this factor operates in opposition to the large round trip times when calculating the bandwidth–delay product, TCP reaches a practical limit when round–trip–delay times approach those on the order of Lunar distance (~2.75 sec). Beyond this distance/time limit, UDP based transports are required.

4.1.4.3 Multiple Ground Stations and Mobile IP

In order to increase the total volume of data transfers without increasing the link data rate, a mission may elect to increase the number of contacts available by using multiple ground stations. For unidirectional UDP–based telemetry downlink, this imposes no additional derived requirements, as each UDP/IP packet has a full destination address and is automatically routed to the correct destination address by the ground network routers. For bi–directional links, utilizing either UDP or TCP, the Mobile IP (MIP) protocol can be utilized to automatically establish a forward routing tunnel from the control center to the spacecraft. This makes it appear that the spacecraft is directly attached to the control center's LAN, regardless of which ground station is in use. MIP is a lightweight protocol, requiring only three packets (advertisement, request, and acknowledgement) to establish the tunnel. Tunnel setup time is approximately 50 ms + 1.5 round–trip–times. Once established, the tunnel is completely transparent to all forward communications, and the data is automatically routed to the correct ground station by the ground network routers. In the event of a contingency, when only a forward link is available, a manual tunnel can be established to the correct ground station in order to support blind commanding. Appendix A.2 provides addition details on Mobile IP

4.2 Space – Space Cross Links

This section will cover the use of IP for communications between satellites in–orbit. It will focus on the concepts of ad–hoc, dynamic networking and routing requirements as nodes (satellites) enter/leave a TBD–identified group of other spacecraft that currently form a local area network. This section focuses on the establishment of the networking routing services for peer–to–peer, or for a many–to–one or a one–to–many, network.

This section will be refined as additional research and work are completed.

SECTION 5 MISSION SECURITY REQUIREMENTS[1]

Security for a mission using an IP-in-space approach is simply a continuation of existing security concepts that are required for any space mission, which will be based upon the concepts as identified in NPG 2810.1 (NASA Procedures and Guidelines; Security of Information Technology) and the corresponding GSFC version (GPG 2810.1). The present document does not supercede the security requirements specified in NPG 2810.1 or in any other official policy document. Section 5.1 and section 5.2 are intended to refamiliarize the SE with the concepts and definitions contained in NPG 2810.1 and the IP-in-Space Handbook. Section 5.3 describes security features related to using IP in a complete end-to-end architecture as well as other non-IP security issues.

It is specifically cautioned that since mission security is an area of on-going study, official policies and guidelines are expected to evolve. System engineers and other responsible project personnel must ultimately follow current versions of official policies and guidelines. The material in the present document regarding security should only be considered as introductory.

5.1 Reference Information from NPG 2810.1

At GSFC, the flight projects are responsible for implementing the guidelines and processes as defined in NPG 2810.1. This document identifies “what” the projects implement. The NPG describes the NASA Information Technology (IT) Security Program, providing directions designed to ensure that safeguards for the protection of the integrity, availability, and confidentiality of IT resources (e.g., data, information, applications, and systems) are integrated into and support the missions of NASA.

Missions contain a variety of information types, and these different types of information travel over different paths in the networks. The path, its security characteristics, and the information type all are considered when determining risk. The guidelines, as defined in NPG 2810.1, specify five information types into which all of the mission information should be categorized; this is recapitulated in Table 5-1.

[1] This section is being revised based on general security discussions. As of this release (September 2003), this is the current baseline for the topic of mission security concepts, which is a subject of increasing study.

Table 5-1 NPG2810.1 Data Informational Categories

NASA Information Category	Data in Category	Impact of Loss
Mission (MSN)	If the information, software applications, or computer systems in this category are altered, destroyed, or unavailable	The impact on NASA could be catastrophic. The result could be the loss of major or unique assets, a threat to human life, or prevention of NASA from preparing or training for a critical Agency mission
Business and Restricted Technology (BRT)	This category consists of information that NASA is required by law to protect It includes information, software applications, or computer systems that support the Agency's business and technological needs.	In general, if information in this category should be disclosed inappropriately, the disclosure could result in damage to employees, loss of business for NASA's partners and customers, contract award protests, or the illegal export of technology
Scientific and Engineering Research (SER)	This category contains information that supports basic research, engineering, and technology development but is less restricted against public disclosure	Alteration, destruction, unauthorized disclosure, or unavailability of the systems, application, or information would have an adverse or severe impact on individual projects, scientists, or engineers; however, recovery would not impede the Agency in accomplishing a primary mission.
Administrative (ADM)	This category includes systems, applications, and information that support NASA's daily activities, such as electronic mail, forms processing, networking, and management reporting	
Public (PUB)	This category contains information, software applications, or computer systems specifically intended for public use or disclosure, such as a public Web site or hands-on demonstrations	The loss, alteration, or unavailability of data in this category would have little direct impact on NASA's missions, but it might expose the Agency to embarrassment, loss of credibility, or public ridicule

To support the requirements of NPG 2810.1, the SE will use the terms defined in Table 5-2 to help ensure mission security and to maintain the operational security on the mission network.

Table 5–2 Definition of Security Terms

Security Term	Definition
Availability	This is the state wherein information, data, and systems are in the place needed by the user, at the proper time, and in the form that the user requests
Data Integrity	This is used to define the ability to ensure that information, the applications processing that information, the information technology systems used to run that information, and the hardware configuration, connectivity, and the status of privilege settings cannot be altered during processing, storage or transmission.
Confidentiality	This is used to denote the holding of sensitive data in confidence such that distribution is limited to those individuals or organizations with an established need to know
Authentication and Non–Repudiation	This defines the validation and confirmation of an IT user's claim of identity, occasionally referred to as personal authentication The validation and identification of a computer network node, transmission, or message.”
Access Control	This identifies the ability to obtain or change information or data. Within a system, "access" is the interaction between a subject (e.g., person, process, or device) and an object (e.g., record, file, program, or devise) that results in the flow of information from one to the other. The nature or type of access can be read, write, execute, append, modify, delete, and create.”
Traffic Flow Integrity	This defines the methods used to prevent the collection of sensitive information about the network through observation of the network traffic characteristics. This includes gaining information about the network based on when traffic does or does not flow or based on packet headers being sent to, from, or within the network

5.2 IP Security Concepts

The NASA/GSFC Enterprise Information Technology Security Branch (EITSB), personnel, with support from the Advanced Architectures and Automation Branch, performed a security analysis review of using IP as the space communications protocol and categorized a set of threats, vulnerabilities, and various control mechanisms that should be reviewed by the system engineer to ensure that data and systems integrity are maintained for the mission. The branch completed this study and documented their results in the IP-in-Space Security Handbook[2] (baselined in September 2001). The results of this study can be found on a GSFC web site[3]:

As defined by the IP-in-Space handbook, any uplink, or communication destined for the spacecraft, including commands, is categorized as Mission (MSN) Information. Some downlink information, or information originating at the spacecraft, will also fall into the MSN category. Examples of this type of information include IP addresses contained in the telemetry data, encryption keys, or digital certificates. If these data are lost, misused, or comprised, it could result in loss of key mission data or systems or even cause the loss of the spacecraft. Any other downlink may be BRT, SER, ADM, or PUB information.

Many security solutions are already widely available for use with IP and many more will be developed in the future. Security solutions need to be tailored to an appropriate level for each mission based on the mission size, the acceptable level of risk, and mission budget, just to name a few. The initial deployment of IP in space will probably use private networks just like the current ones that have been in use since 1998 with missions like UoSAT–12 and the recently launched CHIPSat and CANDOS missions.

[2] The referenced handbook is composed of 3 separate documents: an ops concept, a risk assessment, and a prototype results document

[3] The URL is http://forbin2.gsfc.nasa.gov/297/docs/ip-in-space.stm

5.3 Mission Analysis with an IP Approach

If needed, the SE would first review the information specified in section 5.1 and 5.2. Then the SE will commission a trade study for how his specific mission plans to incorporate IP and what safeguards his mission will use to mitigate the possible risks. The analysis will focus on keeping the systems, data, and networks safe from risks that result from unintentional or malicious threats and vulnerabilities. Regardless of the protocol used, threats and vulnerabilities always will exist to some degree. For each identified threat or vulnerability, the system engineer must chose appropriate control constructs to ensure the viability of the mission.

The IP–in–Space handbook listed several potential solutions resulting from their prototyping activities (Reference Section 3, Test Results Summary from the Technology Recommendations Report). However, the results listed are based upon a simulated spacecraft.

The SE will evaluate the possible solutions that will be used to mitigate the identified security threats and vulnerabilities based on his represented architecture for on–board spacecraft systems and for the complete ground system data transfer. Each solution comes with an associated cost in terms of project funds, reduced CPU utilization, or reduced data throughput rates.

The SE will evaluate risks based of the impact assessment results. The risks are associated with the inadvertent, or malicious, intent to compromise the uplink command data that potentially could result in the following consequences, listed in increasing order of consequence to the mission:

- Minimal, or negligible, impact
- Loss of redundancy (either in mission assets, data, or mission services)
- Loss of science, but might still be able to meet some Project Level 1 requirements
- Loss of all science data
- Loss of spacecraft

In any of these risks, the attacker's intent is either to cause a denial of service, to cause the system or data to become corrupt or useless, or to send either a command (correctly formed or mal-formed) that results in the spacecraft's loss of normal science collection

5.3.1 Recommended Control and Access Requirements

The following is a list of control and access capabilities that can be used to provide the mission with a sufficient level of security; some of these relate to facility security (pass cards and biometrics); others support systems security (stateful firewalls, packet filtering, and intrusion detection systems); some ensure software and systems security (encryption, SSL, IPSec, Key management); while others should be included as standard practices (NPG 2801.1 compliance, education of users, code testing)

Missions may use the following information to help determine the security architecture they need to employ in their mission. This information will provide proof–of–concept solutions. However, it will **not** provide a perfect solution for securing IP–in–Space communications; perfect solutions are not possible. The SE should tailor the use of several of the following control and access requirements to ensure the integrity of both the mission data and the supporting systems.

The following table provides a brief overview on the various controls that the SE can use to provide for a complete security approach. This table is derived from the information contained in the IP-in-Space Security Handbook, Risk Assessment and Information Protection Recommendations, Appendix C.

Table 5–3 Sample Security Control Concepts

Access Control Method	Access Control Concepts	Access Control Results
Proof of NPG 2810.1 Compliance	This plan should outline ALL measures taken to secure the network, devices, and facility to the level deemed necessary by NPG 2810.1	Audited and approved Security Risk Mitigation Plan (by EITSB) ensures no possible unauthorized access and that system, network, and data are secure; can not be compromised, or made XXX
Education of Users	Users who have had security education are less likely to make a mistake that will lead to a security breach, and are more likely to accept additional process steps necessary to maintain security	Ensures that all users are aware of control concepts and preventive actions such as physical security, password protections and other common, overlooked aspects of IT security
Physical Security	Physical security should be strictly maintained on mission networks. All mission network equipment should be behind locked doors, and only authorized necessary personnel should be granted access	Prevents unauthorized access to secure areas; no unauthorized personnel can have access to command and control (C&C) components o sensitive data that could be used to defeat other security protocols such as IP addressed, encryption keys, or certificates
Radio Frequency Protocols	The use of frequency hopping or spread spectrum may make attacks on Radio Frequency (RF) communications harder to accomplish	While not specifically IP, use of frequency hopping or spread-spectrum makes attacks on RF links harder to accomplish; minimizes ability of others to “sniff” traffic may not be feasible for smaller projects or those with limited resources
Authentication Devices	Authentication may be based on something you have, something you know, or something you are. By using more than one of these “some things,” such as a keycard that you have and a password that you know, authentication may be made more reliable	This keeps unauthorized personnel from secure areas; prevents malicious attacks via hackers from gaining access into secure areas
Passwords	Strong passwords can be used to help prevent false authentication and subsequent access compromise. Strong passwords should not be words found in any language dictionary, should contain at least 8 characters, and should be a combination of upper case letters, lower case letters, special characters, and numbers. An even better solution is the use of “One–time passwords”. These are preferred to strong passwords when the password must be sent in the clear over an accessible link	Prevents unauthorized access to secure C&C systems; prevents hackers from gaining access to systems
Smartcards	Smartcards usually are used in conjunction with a password. A smartcard will either hold a chip that is recognized and approved by a card reader, or generate a one–time password.	Same as above
Biometrics	Biometrics is also a form of two–fold authentication. Users use a password (something you know) and a biometric signature, such as a fingerprint, facial image, iris scan, or voice print (something you are) to gain authorization.	Examples of DoD-like security with fingerprints, retinal scans; usually not practical or cost-effective for NASA projects unless specifically mandated by project

Access Control Method	Access Control Concepts	Access Control Results
Digital Certificates	Digital certificates are structures digitally signed by one entity carrying the public key and associated information of another entity. There are several standards using digital certificates. The financial industry uses X.968 from the International Telecommunication Union (ITU).	Another authentication concept to ensure data integrity; such as digitally signed commands sent by science center can only be decrypted by MOC; used to ensure instruments not compromised by hacker spoofing the science center
Code Testing and Auditing	Code testing and auditing can promote successful authentication, access control, data integrity, and availability. Ensuring that the code is written well will limit the flaws that can be exploited to circumvent control measures.	Ensures that systems built with IP concepts are fully tested; no backdoors or generic passwords in place. All conditions specifically handled preventing unauthorized personnel from gaining access to critical systems or components; limits flaws that could be exploited to circumvent control features that could be used to prevent systems availability
Packet Filtering	Packet filtering restricts access based on IP address, which is a subset of access control that a strong stateful firewall provides.	Another access control concept; Restricts data from non-standard IP addresses from gaining entry into command center; allows only specific end-points to communicate through firewall
Stateful Firewall	A stateful firewall is a filter with the addition of the ability to track the state of session–oriented connections. The firewall should support logging capability.	Ensures only specified traffic has access over network of secure facility; logging provides concepts of who data can into or out of
Encryption	Encryption is used in many protocols and tools to restrict access to and provide confidentiality for information.	Either HW or SW support; used to restrict access into data fields and provides data confidentiality; does increase cost and complexity and might also result in performance degradation
Key Management	A robust, secure key management system must be developed for any system using encryption.	Used in encryption technology; however, DO NOT LOSE KEYS; otherwise may not be able to communicate between end points; if a key o certificate is lost or compromised, a method already MUST be in place to establish a new key
IPSec	IPSec is the standard IP protocol used to create Virtual Private Networks (VPNs). The protocol creates an encrypted tunnel between two endpoints, which may be either hosts or gateways	With IPSec, all communications between 2 endpoints is encrypted
SSL and TLS	Secure Socket Layer (SSL) and Transport Layer Security (TLS) provide server and client authentication and encryption for a single network connection.	Authorization and encryption for single network connections; this option selectively encrypts those streams based on either port or socket information
Intrusion Detection Systems (IDS)	IDS do not actively provide any security. Instead, they detect when the security controls have been defeated, so that appropriate corrective action may be taken.	IDS detects when a security control has been defeated via unauthorized access; allows for personnel to take corrective action

Any solution should include strong perimeter security, network and host–based security measures, intrusion detection and logging (with appropriate monitoring and reaction plans), strong authentication using multiple types of identifiers, and layers of diverse security measures, so that there is no single point of failure in the security scheme. The following table briefly summarizes the access control capabilities that each mission should consider when developing the overall mission. Again, this information is provided in more detail from the IP–in–Space Security documents located at: http://forbin2.gsfc.nasa.gov/297/docs/ip-in-space.stm

In many cases, system users rely on one or a few strong security controls, such as a VPN or firewall system. However, it must never be assumed that because of a specific control, all risk has been eliminated. Because any given control may be compromised at some point in the future, systems should be built with layers of security. For example, network–based measures such as firewalls and filtering routers should be used with host–based measures such as TCP Wrappers. This will ensure that if an attacker gets past one countermeasure, another will be in place as an additional line of defense.

5.3.2 IP Security Trade Studies

The SE will review several possible solutions that will be used to minimize the security threats and vulnerabilities mentioned in the previous section. Each solution comes with an associated cost in terms of project funds, reduced CPU utilization, and reduced data throughput rates. The IP–in–Space handbook included several examples of these aspects in the results of the prototyping activities (Reference Section 3, Test Results Summary from the Technology Recommendations Report).

However, for the mission that the SE is working, the spacecraft systems will be different from the simulated spacecraft used for that report. The SE must evaluate possible security solutions based on his represented architecture for on–board spacecraft systems and for the complete ground system data transfer. The following list is currently a placeholder and will be completed in future releases of the present document.

- Digital Certificates Trade Study Setups
- Packet Filtering Trade Study Setups
- Data Encryption Trade Study Setups
- Key Management Trade Study Setups
- Secured Socket Layer (SSL) Trade Study Setups
- Virtual Private Network (VPN) Trade Study Setups
- Radio Frequency Trade Study Setups

The IP-in-Space handbook already contains several of these security setups and includes test results as well. The setup conditions as well as the results are found in the IP-in-Space Handbook; IP-in-Space Security technology, Solutions, Recommendations and Prototype Efforts, Appendix A through Appendix I.

Appendix A. Protocol Layering

The following sections provide additional information on the header formats for the various IP protocols. For each of these headers, the user **DOES NOT** provide or develop any user–specific code. These headers are inherent using the RFC compliant IP stack

This section provides a brief IP tutorial and some examples of the physical components required for the ground–ground communications and the space–ground communications using an Internet protocol.

A.1 IP Tutorial

Figure A–1 provides the information related to the seven layers associated with the IP implementation concepts. As depicted in this figure, the layering principle is critical; it provides a clean and separate approach that:

- Isolates special space problems so they can be addressed as needed
- Allows independent implementations
- Provides a modular approach that allows upgrading individual areas

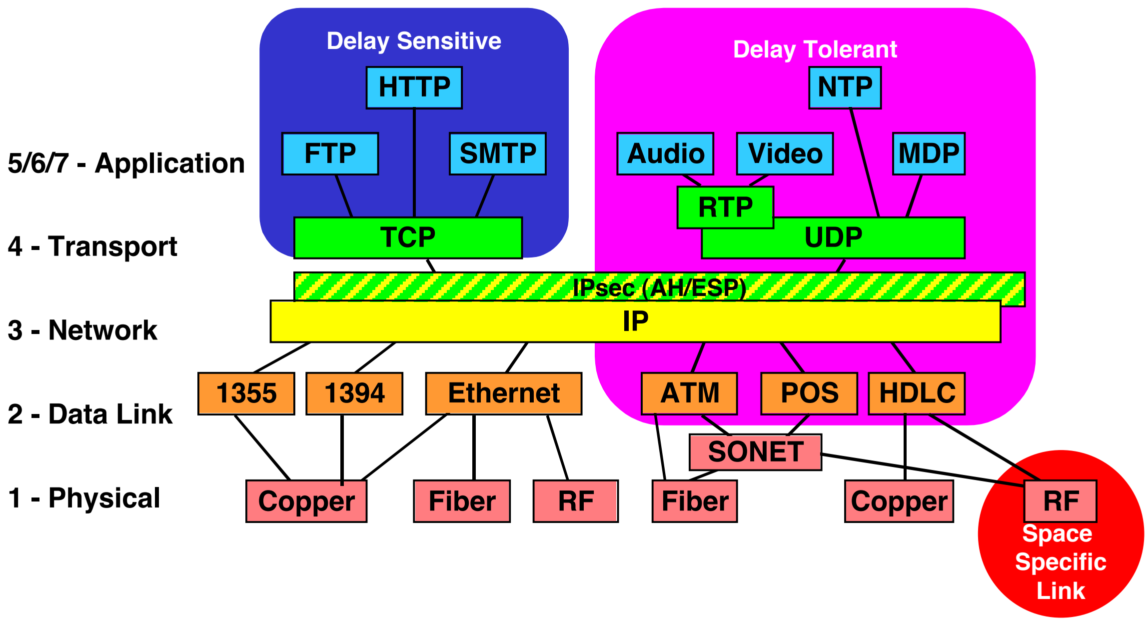

Figure A–1 IP Layering Concepts

A.1.1 Physical Layer

The physical layer is the mechanism for delivering bits across media (e.g., copper, fiber, RF). Various trade studies, such as power, antenna gain, distance, noise, data rate, modulation, and frequency, are performed to determine what media is best suited for a particular application.

The main issue is making space RF (or possibly optical) link deliver the bits across the medium. The RF system must be built for space and is independent of any upper layer protocols.

A.1.2 Data Link Layer

The data link layer splits data into frames, fragmenting as necessary, for sending on the physical layer. The protocol used in the data link layer has a characteristic Maximum Transmission Unit (MTU), which determines whether a network layer packet must be fragmented. Typical MTUs are on the order of 1000 to 2000 bytes. Ethernet, for example, has an MTU of 1500. One of the most important data link control protocols is High–Level Data Link Control (HDLC). Not only is HDLC widely used, but also it is the basis for many other important data link control protocols, which use the same or similar formats and the same mechanisms as employed in HDLC.

The data link layer is responsible for transmit and receive functions over the network. For the transmit function, the data link layer inserts the frames into the upper layer protocol data units over the physical layer, and it adds the error detection to transmitted frames. During the receive function, the data link layer extracts frames from the physical layer and passes them up to the network layer; it also performs error detection on the received frames.

A.1.3 Network Layer

The network layer provides global, end–to–end addressing for each data packet. It determines the routing of packets of data from sender to receiver via the data link layer and is used by the transport layer. The most common network layer protocol is IP, providing 32 bits of source address and 32 bits of destination address.

The network layer provides automated management of routing tables; it is implemented in routers and end–system operating systems. The network layer is the key to the success of the Internet; it provides a standard, fixed–format protocol header, which is key to global interoperability. If this protocol header is not followed exactly, communicating across the Internet is impossible.

A.1.4 Transport Layer

The transport layer determines how to use the network layer to provide a virtual, point-to-point connection so that host A can send messages to host B. The two transport protocols that are widely used are the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

TCP is connection–oriented and stream–oriented; it provides “reliable” end–to–end data delivery. UDP, on the other hand, is a connectionless protocol; it performs “send–and–forget” data delivery and is similar to all current spacecraft frame delivery concepts.

UDP uses a simple header to multiplex user data over IP; there is no session setup or tear down and it works on unidirectional links. UDP is unaffected by propagation delay. If the user needs a “reliable” delivery mechanism within UDP, the user will create a function to provide a feedback loop to ensure the reliable delivery of data. UDP provides an Internet interface that operates similar to traditional spacecraft communication systems that are currently used within NASA.

TCP provides the same multiplexing features as UDP; however, it has additional fields to support “reliable” data delivery. TCP uses sequence numbered datagrams and acknowledgements to provide flow control in response to network performance. TCP is sensitive to a combination of data rate (bandwidth) and delay, and to network errors and congestion. TCP provides a relatively tight feedback loop between end systems to ensure data transport between sender and receiver.

A.1.5 Application Layer

The users will review/determine what the specific application will use to support the transport protocol best suited to their needs (e.g., UDP or TCP). There are numerous standard applications that are available for file transfer, store–and–forward delivery, time synchronization, and non–data formats (audio, video). Additionally, users can develop their own applications to meet special needs.

A.2 Mobile IP Tutorial

Mobile IP, (RFC 2002), is a standard proposed by a working group within the Internet Engineering Task Force; it is designed to solve the problem of IP addressing by allowing the mobile node to use two IP addresses: a fixed home address and a care–of–address that changes at each new point of attachment.

A.2.1 How Mobile IP Works

In today's spacecraft communications, control centers normally send commands to the ground stations/antenna for which the spacecraft is passing over and the command is uplinked to the spacecraft. The major issue is that the control center must know where to send the commands and address the commands accordingly. However, as large constellations of spacecraft are deployed, advance planning and scheduling of contacts becomes more complex and expensive and an automate solution for delivering commands to the proper ground station is desirable.

Figure A-2 shows an example where a spacecraft has an Internet address (e.g. 100.10.10.18), that is part of a ground based subnet (e.g. 100.10.10.x). Any IP datagrams addressed to the spacecraft address from anywhere on the Internet will be routed using standard Internet routing and will be delivered to the Home Agent ground subnet. However, the Home Agent router needs to know how to forward the packet to the proper ground station to get it to its final destination. This is exactly the same problem encountered by other mobile devices on the Internet such as laptops, PDAs, and eventually automobiles

The Internet Engineering Task Force (IETF) has developed standards called Mobile IP (RFC 3220) to deal with this problem. These protocols use an initial protocol exchange to allow the mobile device or spacecraft to determine if it is in direct contact with its home subnet and associated home agent software or a foreign subnet and its foreign agent software. If the mobile device is in contact with a foreign subnet, the foreign agent establishes an IP encapsulation tunnel from the home agent to the foreign agent. Then when the control center sends a datagram to the spacecraft address, the packet goes to the home router where the home agent notices that there is a tunnel to the spacecraft via a foreign router. The packet is then sent through the tunnel to the foreign agent, which passes it out its serial interface and up to the spacecraft.

This sort of Mobile IP scenario is primarily an issue for sending data to the spacecraft. When any packets are sent from the spacecraft to any ground station, the ground station simply uses the destination address to forward the packets using standard Internet routing rules. One possible exception is if the foreign ground station has additional routing rules, for security reasons, which prevent it from forwarding packets whose source address is not within the foreign subnet. Then the tunneling features of Mobile IP would be needed to encapsulate the spacecraft packets for delivery to their home subnet.

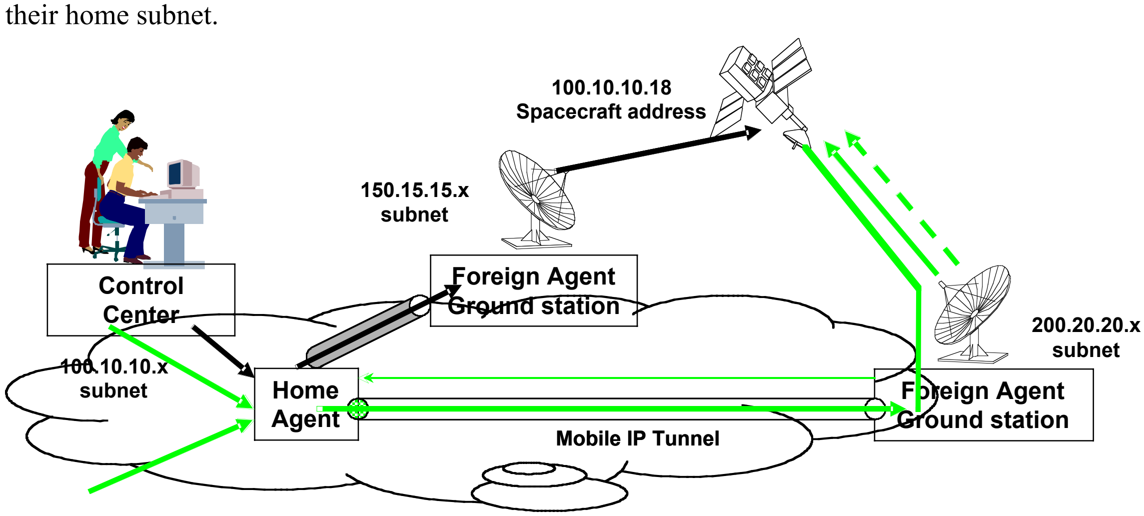

Figure A–2. Mobile IP Communications

These cases have only addressed a spacecraft or mobile host with a single IP address. If the mobile device or spacecraft has a LAN with multiple IP addresses then the problem gets more complex. One solution is for each node on the spacecraft with an IP address to perform Mobile IP registration and set up tunnels for each. However, this does not scale well and causes additional traffic for all of the registrations and additional software for each node. The solution currently being worked on in the IETF is called Mobile Routing. It involves a router that performs all of the Mobile IP operations and none of the nodes on the LAN even realize they are mobile. They simply operate just like they do on a fixed LAN. The research and development in this area is being driven by concepts in which all future automobiles will have onboard LANs with Internet addresses and full mobile Internet connectivity. The size of the automobile market, a potential market for mobile routers, is huge and the commercial research and development investments are substantial. A version of the Mobile Routing protocol is currently available in Cisco routers in version12.2.4 of the IOS software

During the CANDOS mission, Mobile IP operations were carefully monitored to analyze their performance during real space mission conditions. All Mobile IP packets were captured and router Mobile IP authentication and tunnel management activities were logged. The time stamps on the data were compared to the Shuttle position and antenna orientation to understand overall protocol performance especially around the beginning and end of the communication contacts. Section 2.5 provides additional details and information on the Mobile IP concepts and the CANDOS lessons learned of using Mobile IP.

To support the use of mobile IP, the router located at the station must have both mobile IP and IP security protocols enabled. The establishment of the forward link with the spacecraft is supported by a TCP/IP connection. The router will act as the foreign agent and advertise its availability; this agent advertisement is scheduled to occur several seconds before the actual time that the forward link is scheduled to begin. The spacecraft will respond to this advertisement and return authorization packets, which are routed to the MOC for authentication. Within a matter of seconds and with a minimum of 2–3 packets, the spacecraft and the control center have established a tunnel by which data from the control center (*home agent*) can be uplinked to the spacecraft (*mobile node*), via the ground station, in which the router acts as the *foreign agent*.

A.3 Mobile IP Concepts

The following sections are used to show representative samples and examples of how a mission would set up for the use of mobile IP. Mobile IP is essentially a three–step process.

- The foreign agent (the router located at a ground station) periodically transmits a “registration advertisement” which is sent to the RF link and is then broadcast.
  - This periodic advertisement is user–configurable to as frequently as 3 seconds.
  - The advertisement is broadcast over an “available” RF link; if the link is not active, no advertisements are transmitted into space.
  - If the RF link is up, the advertisements are sent on the corresponding frequency. If a router supports several antennas, then multiple advertisements could be set up with different spacecraft, all using the specific RF signals. This would imply that mobile IP could be set up between multiple mobile nodes (the spacecraft) and their destinations (MOCs, SOCs, other s/c).
- The mobile node (the mission spacecraft) “hears” the advertisement request and responds back with a “registration request”, which identifies the home agent (normally, the mission operations center IP address)
- The foreign agent sends the registration request back to the home agent; the home agent will authorize, or deny, the registration request. If the request is authorized, then an IP tunnel is created between the MOC and the ground station for the duration of this station contact.

- After a successful registration request, the "foreign agent" periodically performs another registration advertisement with the mobile node to keep the tunnel between the mobile node and home agent active
  - The agent advertisement identified in step 1 has a finite duration, which is user configurable. The default is 600 seconds (10 minutes); it is configurable between four (4) seconds and 1800 seconds. The current practice is to have this duration set to ***TBS*** minutes. The "foreign agent", the router, will generate a new advertisement request at this pre–defined interval.
  - The foreign agent maintains a "lifetime" timer for each mobile IP connection. If the mobile node and home agent do not re–register within the "lifetime" period, the foreign agent removes all information related to this specific instance of a mobile IP connection between the spacecraft and the MOC.

N.B. → Currently the following subsections are a work in progress, to be updated in a future release.

A.3.1 Setup of the Foreign Agent and Services

This implies the router that exists at each and every ground station that is used to support the mission. The mission engineer should ensure that this setup is available for launch and early orbit support as well as the standard normal operations phases of the mission.

Configure Visitor List

Enable Mobile IP using either IPv4 or IPv6

Generate and send "agent advertisements" by the mission routers

Authenticate and coordinate with home agent

Accept, or deny, registration requests

Configure the "care–of" address list

Tunnel to care–of address using IP encapsulation (IP within IP)

Forward replies to the Mobile Node

Ensure security (access, encryption, etc.) of foreign agent(s) involved in the use of Mobile IP

A.3.2 Setup of Home Agent and Services

Authenticate registration request

Generate registration replies

Accept, or deny, of registration requests

Forward replies to Mobile Node

Ensure security (access, encryption, etc.) of home agent involved in the use of Mobile IP

A.3.3 Setup of Mobile Node and Services

Response to "Foreign Agent" Agent Advertisement; initial attempt

Ensure security (access, encryption, etc.) of mobile node involved in the use of Mobile IP

Transfer of data to/from foreign agent

A.4 Network Layer Protocol

The Network Layer Protocol (NLP) provides a fixed format protocol header. The standard, fixed format header is the key to global interoperability; IP hides the details of the data link layers from the upper layer protocols. This protocol is defined in RFC 1377; a sample layout of the NLP header is shown in Figure A–2.

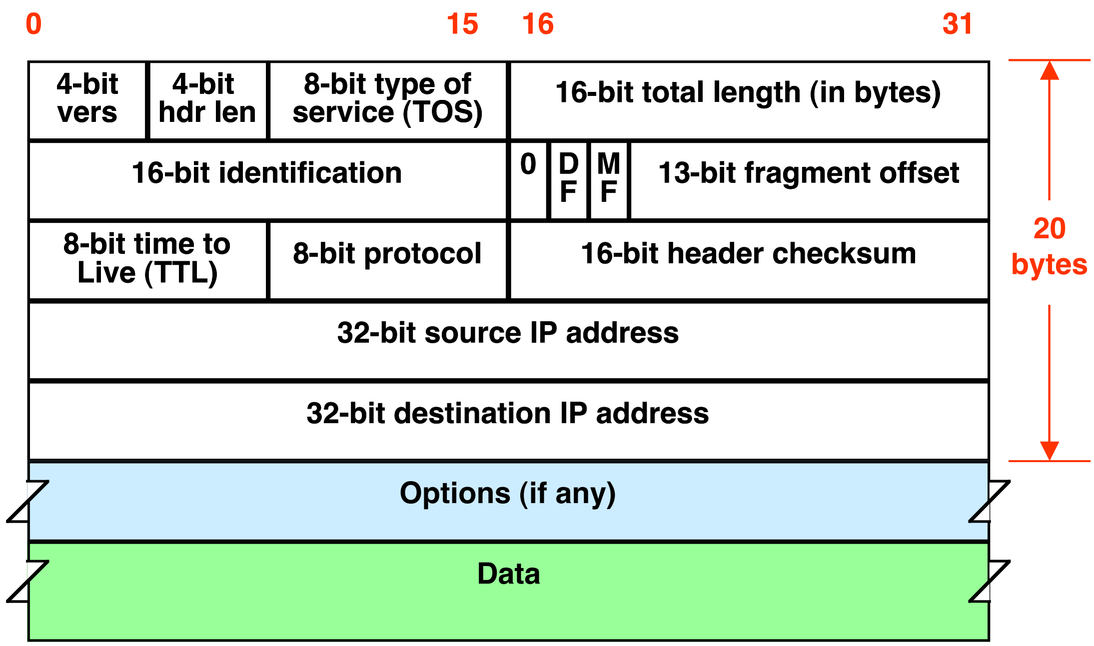

Figure A–3 Network Layer Header Layout

A.5 Transmission Control Protocol

The Transmission Control Protocol (TCP) provides the same multiplexing features as UDP, but it uses additional fields to support "reliable" data delivery. TCP uses sequence numbered datagrams and acknowledgements to coordinate the delivery of data from the source to the destination. It provides flow control in response to network performance. It is sensitive to combination of data rate (bandwidth) and delay; this corresponds to a sensitivity to network errors and congestion. TCP provides a relatively tight feedback loop between end–systems. This protocol is defined in RFC 793, Transmission Control Protocol; a sample layout of the TCP header is shown in Figure A–4.

This RFC describes the functions to be performed by the Transmission Control Protocol, the program that implements it and its interface to programs or users that require its services.

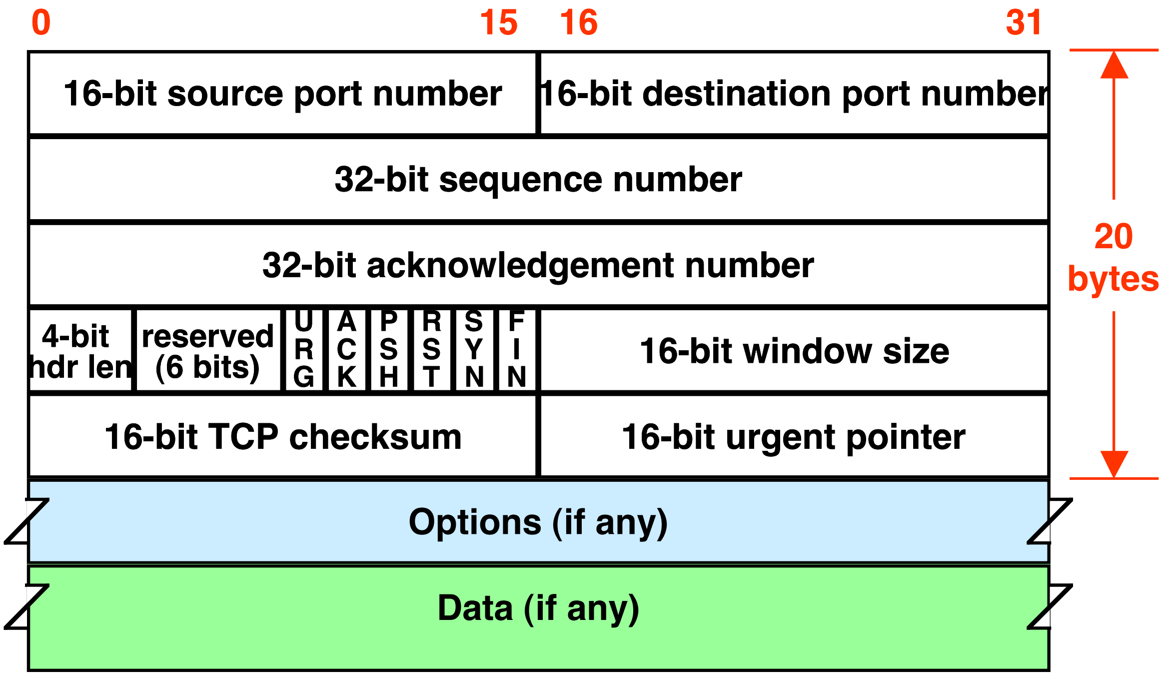

Figure A–4 TCP Header Layout

A.6 User Datagram Protocol

The User Datagram Protocol (UDP) creates a simple header to multiplex user data over IP. There are no session setup or tear down processes required. This protocol works on unidirectional links, unaffected by propagation delay. This protocol is defined in RFC 768, User Datagram Protocol; a sample layout of the UDP header is shown in Figure A–5. This RFC defines the procedure for application programs to send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed.

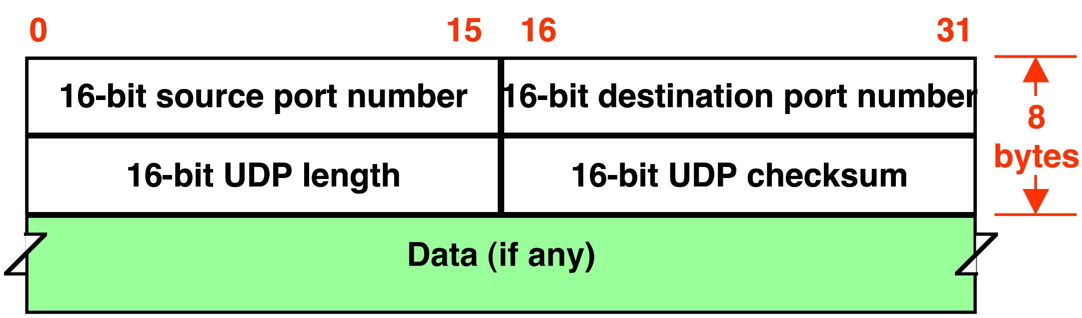

Figure A–5 UDP Header Layout

A.7 Real–Time Protocol

Real–time Protocol (RTP) provides support for reconstructing real–time data streams over UDP. This protocol is defined in RFC 1889; a sample layout of the RTP header is shown in Figure A–6. RTP provides end–to–end network transport functions suitable for applications transmitting real–time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality–of–service for real–time services.

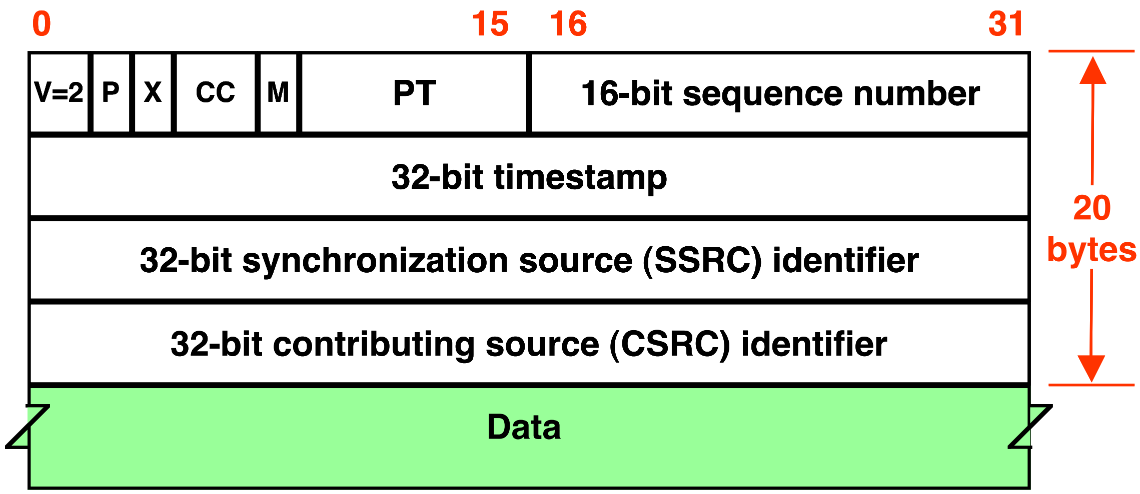

Figure A–6 Real–time Header Layout

A.8 Network Time Protocol

The network time protocol is defined in RFC 1305, Network Time Protocol (Version 3), Specification and Implementation.

Since NTP timestamps represent the main product of the protocol, a special timestamp format has been established. NTP timestamps are represented as a 64–bit unsigned fixed–point number, in seconds relative to 0h on 1 January 1900. The integer part is in the first 32 bits and the fraction part in the last 32 bits.

This format allows convenient multiple–precision arithmetic and conversion to Time Protocol representation (seconds), but does complicate the conversion to ICMP Timestamp message representation (milliseconds).

The precision of this representation is about 200 picoseconds, which should be adequate for even the most exotic requirements.

A.9 Further Refinement of QoS Concepts

In the early days of the Internet, various quality-of-service mechanisms were identified but there were many approaches and not a large amount of interest in implementing them. There is now much greater interest in the Internet community in defining and deploying widely available quality of service mechanisms. The IP header has always has its Type of Service (TOS) field which provides a way to tag packets with this information. However, the exact definition and use of the TOS field has not occurred. However, just tagging packets at the IP level is meaningless unless the lower layers (e.g. Ethernet, Frame Relay, and ATM) can actually support quality of service and are integrated with the IP TOS field.

Network applications such as voice and video over IP and the growing use of IP networking in industrial automation have created a need for better QoS support and standards organizations are working on solutions. The current approach to better QoS in LAN environments is to engineer the LAN to avoid major traffic bottlenecks. This can be done by replacing LAN hubs with switches, which are more selective in passing traffic and avoid sending all packets to all nodes. Another approach is to configure Virtual LANs (VLANs) to segregate traffic flows. These approaches work to reduce congestion and provide better QoS in a LAN environment but they may not provide a solution for users with very strict timing requirements.

Providing better QoS requires interaction between the network protocols and network hardware. Work is underway in organizations such as the IETF and IEEE to develop the protocols and implementation details to provide better QoS. The IETF QoS work includes the following RFCs:

- 2212 Specification of Guaranteed Quality of Service. S. Shenker, C. Partridge, R. Guerin. September 1997.
- 2990 Next Steps for the IP QoS Architecture. G. Huston. November 2000.
- 3317 Differentiated Services Quality of Service Policy Information Base. K. Chan, R. Sahita, S. Hahn, K. McCloghrie. March 2003.
- 3387 Considerations from the Service Management Research Group (SMRG) on Quality of Service (QoS) in the IP Network. M. Eder, H. Chaskar, S. Nag. September 2002.
- 3583 Requirements of a Quality of Service (QoS) Solution for Mobile IP. H. Chaskar, Ed. September 2003.

There is also work on large scale QoS using Multi-Protocol Label Switching (MPLS). Some of the RFCs include:

- 2702 Requirements for Traffic Engineering Over MPLS. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus. September 1999.
- 2917 A Core MPLS IP VPN Architecture. K. Muthukrishnan, A. Malis. September 2000.
- 3032 MPLS Label Stack Encoding. E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta. January 2001.
- 3270 Multi-Protocol Label Switching (MPLS) Support of Differentiated Services. F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, J. Heinanen. May 2002.
- 3353 Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment. D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, F. Ansari. August 2002.

Appendix B. Space–to–Ground Data Link Layer Protocols

The following subsections identify the two of the commonly used data link layer protocols for space–to–ground communications. This link layer protocol is the second layer in the standard Open Systems Interconnect (OSI) model.

B.1 CCSDS Link Layer Protocol

The CCSDS rationale for the space data link layer protocols is defined in Consultative Committee for Space Data Systems, REPORT CONCERNING SPACE DATA SYSTEM STANDARDS, OVERVIEW OF SPACE LINK PROTOCOLS; CCSDS 130.0–G–1, GREEN BOOK, June 2001. This Report provides an architectural overview of the space link protocols recommended by CCSDS and shows how these protocols are used in space mission data systems.

The CCSDS organization has defined several standards for this layer; these are the Packet Telemetry (TM), TeleCommand (TC), and the Advanced Orbiting Systems (AOS). The AOS added on to the TM portion to support the transmission of online data, such as audio and video data.

The CCSDS Recommendations and Reports specific for the Telemetry Systems is defined by CCSDS 102.0–B–5. *Packet Telemetry.* Blue Book. Issue 5. November 2000. The CCSDS Recommendations and Reports specific for the Telecommand is defined by CCSDS 201.0–B–3. *Telecommand Part 1—Channel Service*. Blue Book. Issue 3. June 2000. The CCSDS Recommendations and Reports specific for the AOS is defined by CCSDS 701.0–B–3. *Advanced Orbiting Systems, Networks and Data Links: Architectural Specification*. Blue Book. Issue 3. June 2001.

More recently, CCSDS developed a Proximity–1 Space Link Protocol for bi–directional use over short range, which can support fixed or mobile radio links. This protocol is used to communicate among fixed probes, landers, rovers, orbiting constellations and orbiting relays. This protocol defines both the data link protocol and radio frequency (RF) and modulation characteristics. The CCSDS Recommendations and Reports specific for the Proximity–1 Space Link Protocol is defined by CCSDS 211.0–B–1 *PROXIMITY–1 SPACE LINK PROTOCOL* Blue Book October 2002

Currently, over 200 space missions have used or are planning to use the CCSDS Data Link Layer Protocols.

B.2 HDLC Link Layer Protocol

One of the most common layer 2 protocols is the HDLC protocol. HDLC is a protocol developed by the International Organization for Standardization (ISO). It falls under the ISO standards ISO 3309 and ISO 4335. The protocol uses the services of a physical layer, and provides either a best effort or reliable communications path between the transmitter and receiver (i.e., with acknowledged data transfer).

Several benefits of HDLC are that the control information is always in the same position, and specific bit patterns used for control differ dramatically from those in representing data, which reduces the chance of errors. In HDLC, data is organized into a unit (called a *frame*) and sent across a network to a destination that verifies its successful arrival. The HDLC protocol also manages the flow or pacing at which data is sent. HDLC is one of the most commonly used protocols in what is Layer 2 of the industry communication reference model called Open Systems Interconnection. On transfer of data by the sending entity, programming in layer 3 creates a frame that usually contains source and destination network addresses. HDLC (layer 2) encapsulates the layer 3 frame, adding data link control information to a new, larger frame.

Within the last 20+ years, a wide variety of missions have either been launched or are in development using the HDLC Link Layer protocol. Seventy–two (72) missions from twenty–four (24) different countries, including the USA, Great Britain, Canada, Germany, and France have used this newer protocol.

B.3 Combination of CCSDS coding and HDLC framing

The CCSDS convolutional, Reed/Solomon, and BCH coding mechanisms have been used to provide forward-error-correction on space links for many years. Commercial satellite communication systems that provide Internet connectivity have also used convolutional and Reed/Solomon codes. They both use the same convolutional coding mechanisms but the CCSDS recommendatations specify a more robust code with a little more overhead that can provide better protection over noisy space links.

With CCSDS the R/S coding provides both the code block framing as well as the data framing. In the commercial world, HDLC framing operates completely independent of any framing used in the coding layer. The following diagram shows how variable length HDLC frames can be integrated with the CCSDS recommendations. The variable length HDLC frames are carried over fixed length R/S frames by simply treating the HDLC frames as a bitstream and using the R/S coding as only a mechanism for providing forward-error-correction of a bitstream over the space link. It also show that HDLC can be used without any lower level coding since it provides its own framing mechanism.

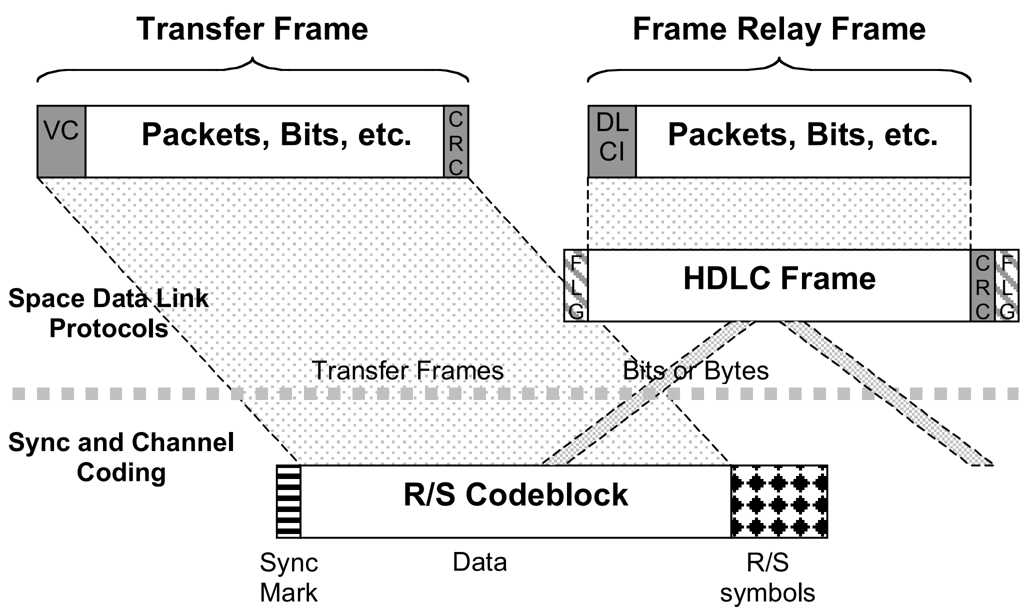

Figure B-2 Integration of CCSDS Coding and HDLC Framing

Appendix C. IP Performance Analysis

C.1 Current Practice Using CCSDS Protocols

Under CCSDS protocols, telemetry data is organized into different kinds of packets, each kind identified by an ApID in the packet header. Various packet options exist, some with an optional secondary header, but all have a minimum of a 6–byte header. In addition to the ApID and length, this header contains some packet sequence information, which maps into the transport layer. Multiple packets are collected into a transfer frame. Packets do not align synchronously within the transfer frame; refer to Figure C–1. Transfer frames are usually a fixed size, which is driven by the Reed–Solomon coding used. For the (255,223) Reed Solomon code specified in CCSDS 101.0–B–4: "Telemetry Channel Coding", this length is an integer multiple of 223 bytes, with a maximum size of 1115 bytes. A transfer frame has mandatory 8–byte header, which combines the functions of the transport, network, and link layers. Some transfer frames contain an optional secondary header, which we will ignore for purposes of this analysis. Finally, the transfer frame is synchronously packed into a Reed–Solomon coding block with a single 4–byte sync pattern and a 32–byte trailer of Reed–Solomon symbols per every 223 bytes of data. Since the Reed–Solomon (and any other coding) used will be identical for either CCSDS or IP, its overhead in each case will be the same (approx. 14%) and can be ignored. Thus, the overhead, excluding R/S, can be calculated for the best case as follows:

n = number of data bytes per packet
t = maximum total bytes per transfer frame = 1115
p = number of packets per transfer frame = (t - 8) / (n + 6)
o = overhead bytes per transfer frame = 8 + 6p
percent overhead = 100 • (o / t)
= 100 • [8 + 6((t - 8) / (n + 6))] / t
= 100 • [0.0071748 + (5.9569506 / (n + 6))]

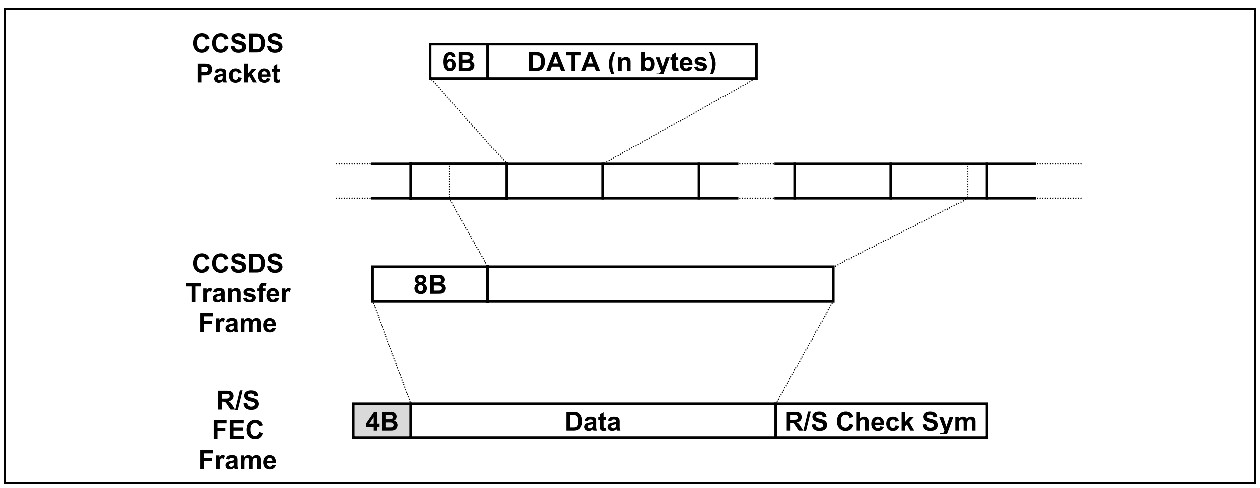

Figure C–1 CCSDS Packet/Frame Setup

C.2 TCP/IP/HDLC Usage

Under TCP/IP, data at the application layer is sent as a reliable stream of bytes. For equality of comparison with CCSDS protocols, it will be assumed that telemetry data is organized into n–byte packets with an equivalent, if not identical, 6–byte header; refer to Figure C–2. At the transport layer, one or more whole application packets are synchronously collected into a TCP packet with a 20–byte header. For purposes of this analysis it will be assumed that there is one application packet per TCP packet. This is a worst–case choice for overhead, but was chosen to compare against the best–case analysis done for CCSDS in section 3.2. At the network layer, each TCP packet is put into an IP packet with a 20–byte header. At the link layer, each IP packet is put into an HDLC frame with a 1–byte sync, a 4–byte header, and a 2–byte trailer. Finally, the HDLC frames are aggregated asynchronously into 1115–byte Reed–Solomon coding block with a single 4 byte sync pattern and a 32 byte trailer of Reed–Solomon symbols per every 223 bytes of data. This is identical to the coding used in the analysis of CCSDS in section C.1. Thus, the overhead, excluding R/S, can be calculated for the worst case as follows:

n = number of data bytes per packet
o = overhead bytes per HDLC frame = 7 + 20 + 20 + 6 = 53
percent overhead = 100 • [o / (o + n)]
= 100 • [53 / (53 + n)]

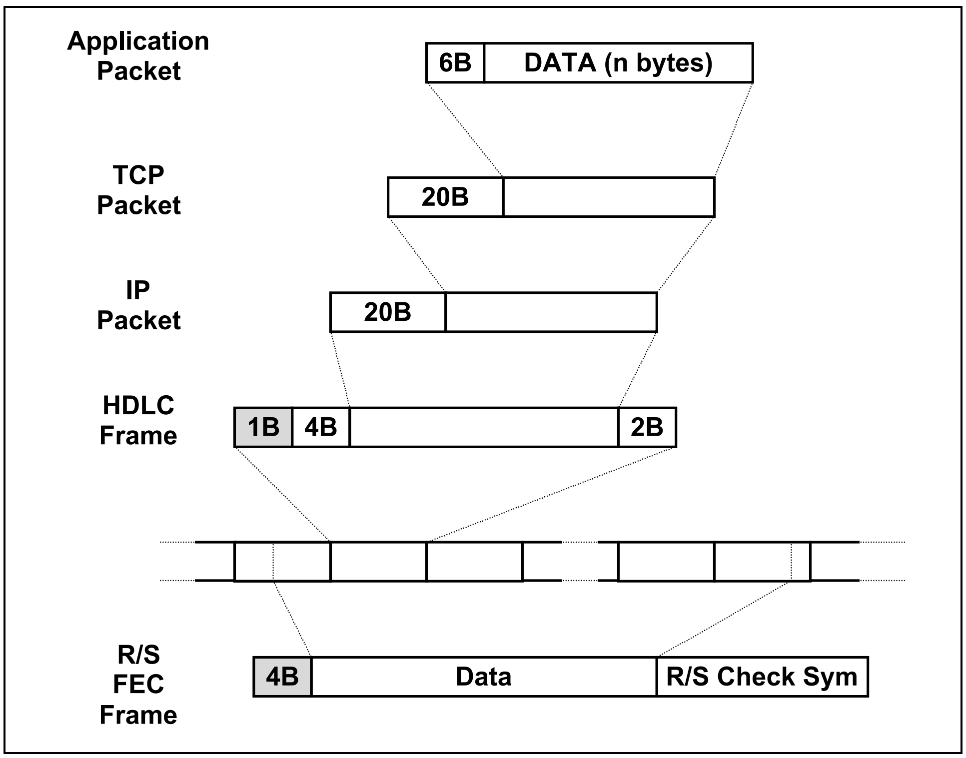

Figure C–2 IP Packet/Frame Setup (With TCP and HDLC Options)

C.3 UDP/IP/HDLC Usage

Under UDP/IP, data at the application layer is sent as an atomic datagram without guaranteed order or delivery. This is functionally equivalent to CCSDS packets. For equality of comparison with CCSDS protocols, it will be assumed that each datagram of telemetry data is organized as an n–byte packet with an equivalent, if not identical, 6–byte header. See Figure C–3. At the transport layer, each application packet is placed into a UDP packet with an 8–byte header. At the network layer, each TCP packet is put into an IP packet with a 20–byte header. At the link layer, each IP packet is put into an HDLC frame with a 1–byte sync, a 4–byte header, and a 2–byte trailer. Finally, the HDLC frames are aggregated asynchronously into 1115–byte Reed–Solomon coding block with a single 4 byte sync pattern and a 32 byte trailer of Reed–Solomon symbols per every 223 bytes of data. This is identical to the coding used in the analysis of CCSDS in section C.1. Thus, the overhead, excluding R/S, can be calculated as follows:

n = number of data bytes per packet
o = overhead bytes per HDLC frame = 7 + 20 + 8 + 6 = 41
percent overhead = 100 • [o / (o + n)]
= 100 • [41 / (41 + n)]

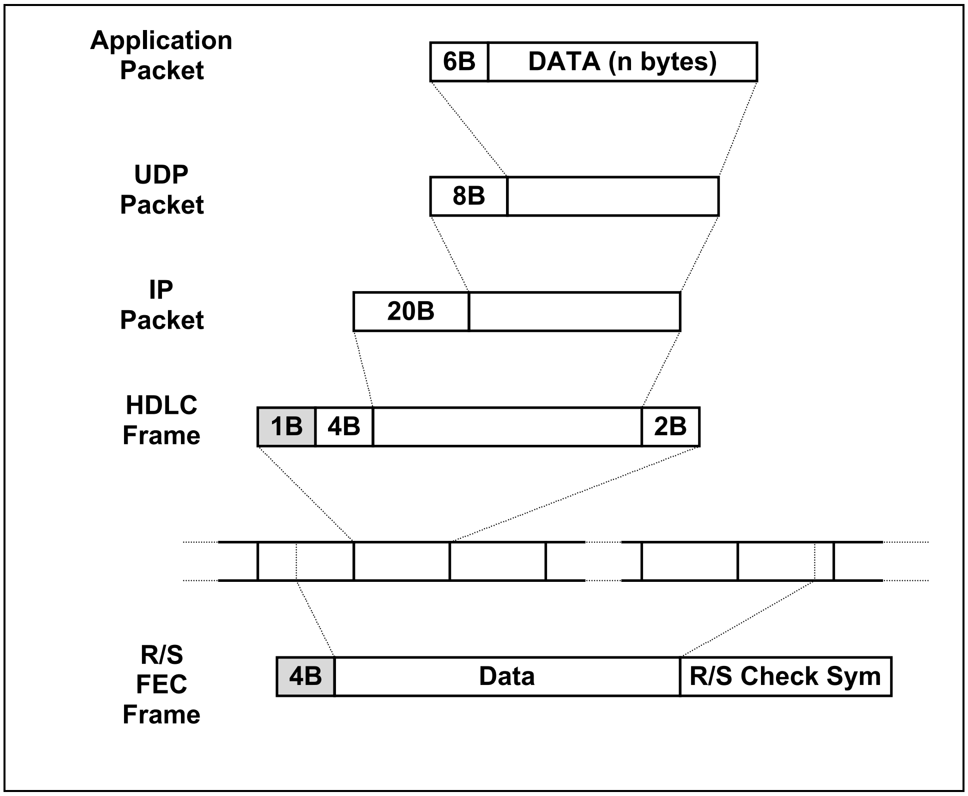

Figure C–3 IP Packet/Frame Setup (With UDP and HDLC Options)

C.4 IP Header Compression

The Internet Engineering Task Force (IETF) is in the process of completing a standard for header compression known as RFC–2507 "IP Header Compression". Headers of typical UDP or TCP packets can be compressed down to 4–7 bytes including the 2–byte UDP or TCP checksum. This largely removes the negative impact of large IP headers and allows efficient use of bandwidth on low and medium speed links. The compression algorithms are specifically designed to work well over links with nontrivial packet–loss rates. Several wireless and modem technologies result in such links.

A typical application of this standard takes the headers for the transport and network layers for either TCP/IP (40 bytes) or UDP/IP (28 bytes) and compresses approx. 90% of them down to 6 bytes.

Thus, the overhead for TCP/IP with header compression is:

percent overhead = 100 • [(0.1•53 + 0.9•17) / (0.1•53 + 0.9 •17 + n)]

The overhead for UDP/IP with header compression is:

percent overhead = 100 • [(0.1•41 + 0.9•17) / (0.1•41 + 0.9 •17 + n)]

C.5 Summary Comparisons

Table C–1 shows the overhead comparison between small (100 byte) and average (1000 byte) packets for all of the protocols analyzed. Figure C–4 shows a graph of all the results for different sized packets. ***Table C–1. Comparison of Header Overhead for Different Protocols***

Protocol	100 Byte Packets	1000 Byte Packets
TCP/IP/HDLC	34.6%	5.0%
UDP/IP/HDLC	29.1%	3.9%
TCP/IP/HDLC w/header comp.	17.1%	2.0%
UDP/IP/HDLC w/header comp.	16.2%	1.9%
CCSDS	6.3%	1.3%

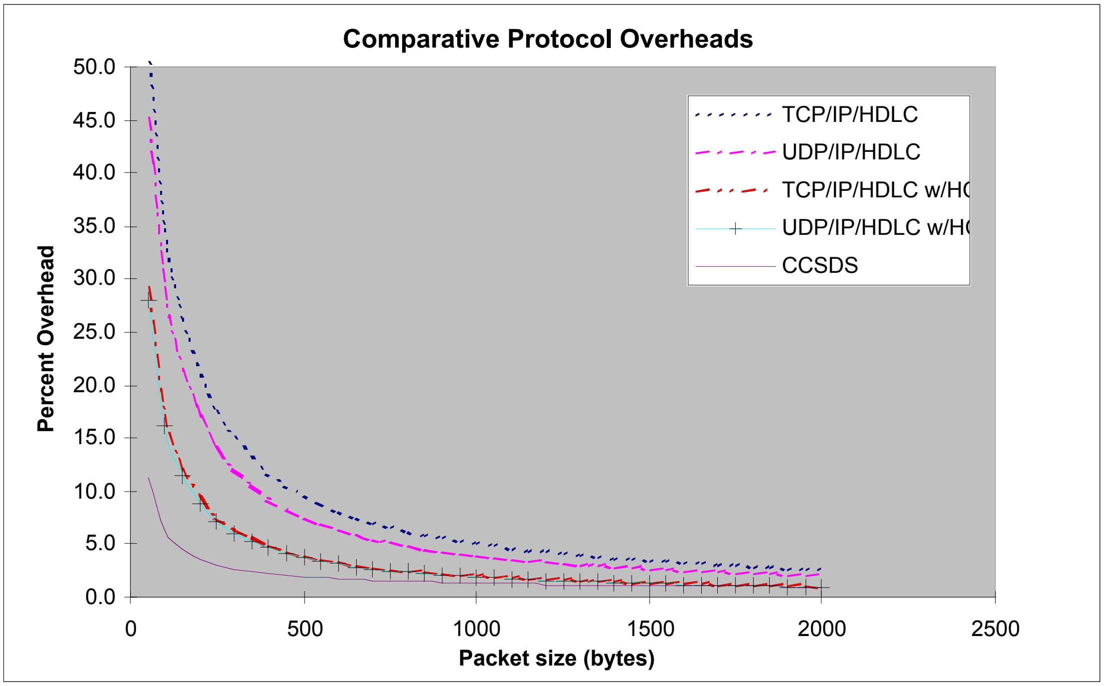

Figure C–4 Comparison of Protocol Overheads

C.6 Conclusions

With its real–time nature and non–guaranteed delivery, CCSDS packet telemetry most closely resembles UDP/IP. For nominal 1000 byte packets, UDP/IP's overhead is 3.9% as compared to CCSDS's 1.3%, a difference of only 2.6%. If UDP/IP with header compression is considered, the difference drops to only 0.6%. These minor differences are insignificant when compared to the Reed–Solomon overhead of 14%.

TCP/IP has a somewhat higher overhead of 5.0%, which is a difference of 3.7% above CCSDS; however, TCP/IP provides additional functionality over CCSDS packet telemetry in the form of automatic retransmission of lost or damaged packets. This will buy additional contact time by allowing operation closer to the horizon where contact becomes intermittent

Appendix D. UDP–Based Reliable File Transfer Protocols

D.1 CCSDS File Delivery Protocol

The CCSDS File Delivery Protocol (CFDP) is a reliable file transfer protocol and is based on user–defined datagrams; one implementation of this datagram could be a UDP–based datagram. This protocol is defined in the "RECOMMENDATION FOR SPACE DATA SYSTEM STANDARDS"; referenced as CCSDS 727.0–B–2 BLUE BOOK; and released in October 2002.

The user can find additional detains on this protocol at http://www.ccsds.org/documents/727x0b2.pdf.

This Recommendation defines a protocol suitable for the transmission of files to and from spacecraft data storage. In addition to the purely file delivery related functions, the protocol also includes file management services to allow control over the storage medium.

The protocol is capable of operating in a wide variety of mission configurations, from relatively simple low earth orbit spacecraft to complex arrangements of orbiters and landers supported by multiple ground facilities and transmission links. In its simplest form, the protocol provides a *Core* file delivery capability operating across a single link. For more complex mission scenarios, the protocol offers *Extended* operation providing store–and–forward functionality across an arbitrary network containing multiple links with disparate availability.

The protocol is independent of the technology used to implement data storage and requires only a few fundamental filestore capabilities in order to operate. It assumes two filestores, one within the spacecraft and one on the ground, and operates by copying data between the two filestore locations.

D.2 Multicast–Dissemination Protocol

The Multicast Dissemination Protocol (MDP) is a protocol framework and software toolkit for reliable multicasting data objects including files and application memory blocks. A primary design goal of MDP is to provide a reliable multicast protocol approach, which is suitable for reliable dissemination of data over both wireless and wired networks.

MDP software has been demonstrated across a range of network architecture and heterogeneous conditions including; the worldwide Internet MBone, bandwidth and routing asymmetric network connections, high delay satellite networks, and mobile radio networks. MDP integrates numerous multicast protocol advances including highly robust, packet–based erasure correction techniques and adaptive group timing mechanisms.

The present MDP software toolkit includes a library with a well–defined API. Several example working applications including a multicast file transfer applications and a very basic multicast chat application are also provided. The user can find additional details on this protocol at http://manimac.itd.nrl.navy.mil/MDP/.

D.3 NACK–Oriented Reliable Multicast

Material in this section is partly adapted from material found at http://norm.pf.itd.nrl.navy.mil/.

The NACK–Oriented Reliable Multicast (NORM) protocol is UDP–based and is currently under development within the Internet Engineering Task Force (IETF) Reliable Multicast Transport (RMT) working group.

The NORM protocol is designed to provide end–to–end reliable transport of bulk data objects or streams over generic IP multicast routing and forwarding services. NORM uses a selective, negative acknowledgement (NACK) mechanism for transport reliability and offers additional protocol mechanisms to conduct reliable multicast sessions with limited "a priori" coordination among senders and receivers. A congestion control scheme is specified to allow the NORM protocol to fairly share available network bandwidth with other transport protocols such as Transmission Control Protocol. It is capable of operating with both reciprocal multicast routing among senders and receivers and with asymmetric connectivity (possibly a unicast return path) from the senders to receivers. The protocol offers a number of features to allow different types of applications or possibly other higher–level transport protocols to utilize its service in different ways. The protocol leverages the use of FEC–based repair and other IETF RMT building blocks in its design.

The reader can find additional details on this protocol at the NORM web site, http://norm.pf.itd.nrl.navy.mil/. This web site maintained by the Naval Research Laboratory (NRL) PROTocol Engineering Advanced Networking (PROTEAN) Research Group. The purpose of this site is to provide information on NORM and provide access to the NORM reference software provided by NRL. The web site will be expanded in the future to contain example NORM applications and NORM software development toolkits based on the NRL reference implementation. The NRL NORM work is heavily based on previous work with MDP , which has similar capabilities.

D.4 Digital Fountain Method

The previous three methods of reliable file delivery are based upon a NACK concept in that only those packets that are not delivered are requested again in a retransmission. The Digital Fountain's technology for data distribution is fundamentally different from the previously mentioned concepts.

The architecture consists of a Digital Fountain server called a Transfer Fountain and Digital Fountain client software called a Fountain Client. The key to the architecture's reliability, high performance, and scalability is the patented concept of Meta–Content. For each piece of content, the Fountain generates a potentially limitless stream of Meta–Content, which consists of mathematical metaphors that describe the original content. Meta–Content has the following fundamental properties:

- Meta–Content packets are independently generated from content at any specified rate—from kilobits per second to megabits per second.
- A bit–for–bit accurate copy of the original content is quickly recovered from any number of Meta–Content packets that in aggregate is equal to the length of the original content.

The second fundamental property implies that packets containing independently generated Meta–Content are completely interchangeable. It does not matter which Meta–Content the Fountain Client receives and in what order. Only the quantity of independently generated Meta–Content received determines when the original content can be reconstructed. Thus, if packets containing Meta–Content are lost in transit, any equal amount of Meta–Content contained in subsequently received packets is just as useful for reconstructing the original content.

More, detailed information pertaining to specific technical details for the Digital Transport technology can he found on their web site, located at http://www.digitalfountain.com/. Please refer to that web site for any additional information relating to Digital Fountain and potential data transport product, solutions, and support.

The Digital Fountain data deliver protocol is a UDP–based solution that can support either multicast or unicast capabilities; this feature is dependant upon whether the recipient can accept the multicast traffic. The basis of this protocol is its Meta–content concepts and the Meta–Content Engine as defined by documentation available at this website.

The web site, http://www.digitalfountain.com/company/customers.cfm provides additional details as to the list of corporate companies, such as media, technology, oil & gas, retail firms and government agencies that use the Digital Fountain family of products to transport their mission critical files.

Appendix E. TCP/IP Characteristics and Limitations

E.1 TCP Window Size

The TCP window size is calculated as the product of the data downlink rate (that the SE identified in the parallel s/c and instrument development phase discussed in Section 2.1) and the transmission round–trip delay time. The following table provides examples of what data the SE needs to define his concepts for downlink rate versus on–board memory allocations based on low–earth orbit (LEO), medium earth orbit (MEO), geo–synchronous orbit (GEO) and several deep space orbits.

Table E–1 provides representative orbital concepts and transmission round–trip delay factors for the various mission classes. The round–trip delay factor is simply calculated for the amount of time it would take a signal (traveling at the speed of light) to transmit data from the source to a destination and back to the source again.

The following table provides an approximate TCP–windows size for several downlink data rates. The TCP–window size is defined in SW at the time the TCP socket is created. It can also be changed after socket creation. If not specified, the default size is 8 Kbytes. The standard TCP window size allows for buffer sizes up to 64 Kbytes. If necessary, a larger value can be chosen as denoted by the use of the TCP–Large Windows option (RFC–1323), which allows for windows up to approximately 1 Gigabyte. The size of the TCP–windows will dictate how much data can be downlinked before having to wait for an acknowledgement. Conversely, the downlink rate would affect how big of a data buffer is needed on–board to support the TCP window.

Table E–1 TCP Window Size and Bandwidth Delay Product

Orbit Classification	Downlink Data Rate (bits per second)	TCP–Window Size or Approximate Bandwidth Delay Product (Kbytes)
Low Earth Orbit (LEO)	2250000	1134
	1250000	630
	900000	453
	125000	63
	23750	11
Medium Earth Orbit (MEO)	2250000	12474
	1250000	6930
	900000	4989
	125000	693
	23750	131
Geosynchronous Earth Orbit (GEO)	2250000	68044
	1250000	37802
	900000	27217
	125000	3780
	23750	718
Deep Space – Mars at Conjunction	23750	1564180
Deep Space – Mars at Opposition	23750	7501680

Orbit Classification	Downlink Data Rate (bits per second)	TCP–Window Size or Approximate Bandwidth Delay Product (Kbytes)
Deep Space – Sun/Earth Lagrange L1 Point	23750	299269
Deep Space – 5 R_e	23750	319
Deep Space – 20 R_e	23750	3817

From the above table, the SE would be able to determine the necessary on–board memory storage allocation required to support the corresponding bandwidth.

The application must set its send and receive socket buffer sizes (at both ends) to at least the bandwidth delay product of the link. The peak bandwidth of the link is typically expressed in bits per seconds (bps); this is the actual downlink data rate from the above table. The round–trip delay can be determined from the Table E–1. For example, in a LEO mission, the round–trip delay is approximately 0.00403 seconds. If the downlink data rate is 125000 bps, this would result in a Bandwidth delay product of 63 Kbytes. The SE would determine that if greater downlink rates are needed, a larger TCP window size is required; the larger TCP window size directly translates into larger memory allocations for the OS.

E.2 TCP Link Asymmetry

Because TCP is a bi–directional, handshaking protocol, it requires a certain amount of acknowledgement traffic in the opposite direction of the data transfer. If insufficient bandwidth exists for these acknowledgements, the data transfer rate will be limited, even if there exists sufficient bandwidth in the data transfer direction. This is known as the TCP link asymmetry requirement. TCP/IP packets utilize a 40–byte header. TCP/IP acknowledgement packets (including headers) are 48 bytes. There is one acknowledgement packet sent for every two data packets sent. Thus, the TCP link asymmetry is a function of the data packet size. For typical data packets, on the order of 1K bytes, the link asymmetry required is roughly 50:1. This relationship between data packet size and link asymmetry requirement is shown in Figure E–1. For ground–to–space TCP data transfers, achieving a sufficient link asymmetry is generally not an issue, as the downlink rates are typically much larger than the uplink rates. For space–to–ground TCP data transfers, however, link asymmetry often becomes the limiting factor.

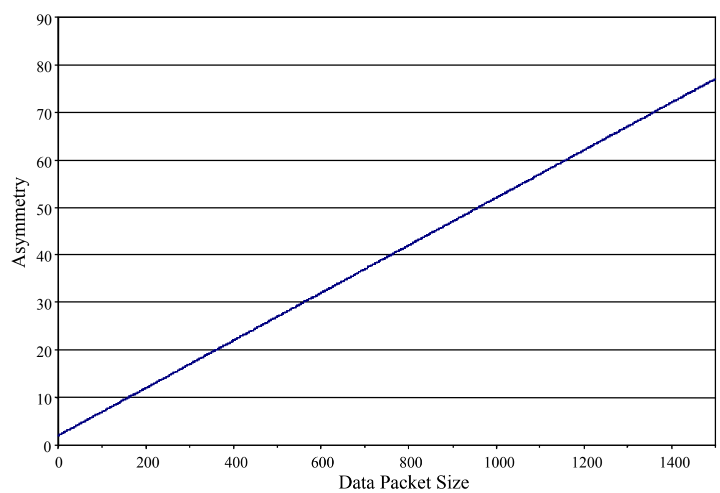

Figure E–1 TCP Data Packet Size vs. Link Asymmetry

Appendix F. Request For Comment (RFC) References

The following RFCs can be found at this web site: http://www.rfc–editor.org/rfc.html.

F.1 RFC Standards

RFC0791 (STD0005)	Internet Protocol, DARPA Internet Program Protocol Specification; [September 1981]
RFC0792 (STD0005)	Internet Control Message Protocol DARPA Internet Program Protocol Specification; J. Postel; [September 1981]
RFC0768 (STD0006)	User Datagram Protocol; J. Postel; [Aug-28-1980]
RFC0793 (STD0007)	Transmission Control Protocol; J. Postel; [Sep-01-1981]
RFC0919 (STD0005)	Broadcasting Internet Datagrams; J.C. Mogul. [Oct-01-1984]
RFC0922 (STD0005)	Broadcasting Internet datagrams in the presence of subnets; J.C. Mogul; [Oct-01-1984]
RFC0950 (STD0005)	Internet Standard Subnetting Procedure; J.C. Mogul, J. Postel; [Aug-01-1985]
RFC0959 (STD0009)	File Transfer Protocol; J. Postel, J.K. Reynolds; [Oct-01-1985]
RFC1157 (STD0015)	Simple Network Management Protocol (SNMP)
RFC2328 (STD0054)	Open Shortest Path First (OSPF) Version 2
RFC2427 (STD0055)	Multiprotocol Interconnect over Frame Relay; C. Brown, A. Malis; [September 1998]

F.2 IP Request for Comments

RFC1256	ICMP Router Discovery Messages; S. Deering; [September 1991]
RFC2002	IP Mobility Support; C. Perkins; [October 1996]
RFC2003	IP Encapsulation within IP; C. Perkins; [May 1996]
RFC2005	Applicability Statement for IP Mobility Support; J. Solomon; [October 1996]
RFC2030	Simple Network Time Protocol (SNTP) Version 4 for IPv4; IPv6 and OSI; D. Mills ; [October 1996]
RFC2125	The PPP Bandwidth Allocation Protocol (BAP), The PPP Bandwidth Allocation Control Protocol (BACP); C. Richards and K. Smith; [March 1997]
RFC2212	Specification of Guaranteed Quality of Service. S. Shenker, C. Partridge, R. Guerin;. [September 1997]
RFC2588	IP Multicast and Firewalls; R. Finlayson; [May 1999]
RFC 2702	Requirements for Traffic Engineering Over MPLS; D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus; [September 1999]
RFC 2917	A Core MPLS IP VPN Architecture; K. Muthukrishnan, A. Malis; [September 2000]
RFC2977	Mobile IP Authentication; Authorization; and Accounting Requirements; S. Glass, T. Hiller, S. Jacobs, C. Perkins; [October 2000]
RFC 2990	Next Steps for the IP QoS Architecture. G. Huston; [November 2000]
RFC 3032	MPLS Label Stack Encoding; E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta; [January 2001]
RFC 3220	IP Mobility Support for IPv4; C. Perkins, Ed.; [January 2002]

RFC 3270	Multi-Protocol Label Switching (MPLS) Support of Differentiated Services; F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, J. Heinanen; [May 2002]
RFC 3317	Differentiated Services Quality of Service Policy Information Base. K. Chan, R. Sahita, S. Hahn, K. McCloghrie; [March 2003]
RFC 3387	Considerations from the Service Management Research Group (SMRG) on Quality of Service (QoS) in the IP Network; M. Eder, H. Chaskar, S. Nag; [September 2002]
RFC 3353	Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment; D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, F. Ansari; [August 2002]
RFC 3583	Requirements of a Quality of Service (QoS) Solution for Mobile IP; H. Chaskar, Ed; [September 2003]

F.3 TCP Request for Comments

RFC0879	TCP maximum segment size and related topics; J. Postel; [Nov 01, 1983]
RFC0896	Congestion control in IP/TCP internetworks; J. Nagle; [Jan 06, 1984]
RFC1106	TCP big window and NAK options; R. Fox; [Jun 01, 1989]
RFC1110	Problem with the TCP big window option; A.M. McKenzie; [Aug 01, 1989]
RFC1180	TCP/IP Tutorial; T.J. Socolofsky, C.J. Kale; [Jan 01, 1991]
RFC1323	TCP Extensions for High Performance; V. Jacobson, R. Braden, D. Borman; [May 1992]
RFC2151	A Primer On Internet and TCP/IP Tools; G. Kessler, S. Shepard; [December 1994]
RFC2398	Some Testing Tools for TCP Implementors; S. Parker, C. Schmechel; [August 1998]
RFC2414	Increasing TCP's Initial Window; M. Allman, S. Floyd, C. Partridge; [September 1998]
RFC2488	Enhancing TCP Over Satellite Channels using Standard Mechanisms; M. Allman, D. Glover, L. Sanchez; [January 1999]
RFC2760	Ongoing TCP Research Related to Satellites; M. Allman, S. Dawkins, D. Glover, J. Griner, D. Tran, T. Henderson, J. Heidemann, J. Touch, H. Kruse, S. Ostermann, K. Scott, J. Semke; [February 2000]

F.4 UDP Request for Comments

3096 Requirements for robust IP/UDP/RTP header compression M. Degermark; [July 2001]

Appendix G. Glossary and Terms

IPsec:

Internet protocol security concepts and options between network and transport layer and also includes the application level encryption. Security solutions will be tailored to the appropriate level for each mission based on mission size, acceptable risk, and mission budget.

Packet:

An efficient application–oriented protocol data unit that facilitates the transfer of source data to users located in space or on Earth.

Protocol:

A set of procedures and their enabling format conventions that define the orderly exchange of information between entities within a given layer of the TM System.

Reed–Solomon ("R–S") Symbol:

A set of J bits that represents an element in the Galois field GF (2^J), the code alphabet of a J–bit Reed–Solomon code.

Reliable:

Meets the quality, quantity, continuity and completeness criteria, without loss of information

Telemetry System:

The end–to–end system of layered data handling services, which exist to enable a spacecraft to send measurement information, in an error–controlled environment, to receiving elements (application processes) in space or on Earth.

Transfer Frame:

A communication oriented protocol data unit that facilitates the transfer of application oriented protocol data units through the space–to–ground link.

Un–Reliable:

Does not meet the quality, quantity, continuity or completeness criteria, which are specified by the TM System.

User:

A human, or machine–intelligent process, which directs and analyzes the progress of a space mission.

Virtual Channel:

This is a sequence of Transfer Frames, which are assigned a common identification code (in the Transfer Frame header), enabling all Transfer Frames who are members of that sequence to be uniquely identified. It allows a technique for multiple source application processes to share the finite capacity of the physical link (i.e., through multiplexing).

Appendix H. Abbreviations and Acronyms

3DES	Triple Data Encryption Standard
8PSK	8-phase shift keying
ADM	Administrative
AH	Authentication Header
API	application programming interface
APID	application process identifier
ASCII	American Standard Code for Information Interchange
ASICS	application–specific integrated circuits
ATM	asynchronous transfer mode
BDP	bandwidth delay product
BER	bite error rate
BRT	Business and Restricted Technology
BSD	Berkeley Software Distribution
BSMTP	batch simple mail transfer protocol (
CCSDS	Consultative Committee for Space Data Systems
COTS	commercial off–the–shelf
CFDP	CCSDS File Delivery Protocol
CPU	central processing unit
CRC	cyclic redundancy code
CSO	Computer Security Officer
DMA	direct memory access
DMC	Disaster Monitoring Constellation
DNS	Domain Name System
DoD	Department of Defense
DVB	digital video broadcasting
EIA-IS	Electronic Industries Alliance – Information Series
EITSB	Enterprise Information Technology Security Branch
ESA	European Space Agency
ESP	Encapsulation Security Payload
FA	foreign agent
FDDI	fiber distributed data interface
FEC	forward error correction
FIPS	Federal Information Processing Standards
FOT	Flight Operations Team
FPGA	field-programmable gate array

FTP	File Transfer Protocol
GN	Ground Network
GPM	Global Precipitation Measurement
GPS	global positioning service
GRID	ground station router interface device
GSFC	Goddard Space Flight Center
HA	home agent
HW	hardware
HDLC	High–Level Data Link Control
HTTP	Hyper Text Transfer Protocol
I&T	integration and test
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Systems
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IONet	Internet Protocol Operational Network
IP	Internet Protocol
IPSec	Internet Protocol Security
iSCSI	Internet Small Computer System Interface
ISL	Inter–Satellite Link
ISO	International Organization for Standardization
IT	Information Technology
ITAR	International Traffic in Arms Regulations
ITU	International Telecommunication
ITT	International Telegraph and Telephone
JPL	Jet Propulsion Laboratory
LAN	local area network
LDPC	low-density parity check
LPT	low power transceiver
MB	Megabyte
MDP	Multicast Dissemination Protocol
MHz	Mega–Hertz
MIL–STD	Military Standard
MTU	Maximum Transmission Unit
MOC	Mission Operations Center
MSN	Mission
NACK	negative acknowledgement

NASA	National Aeronautics & Space Administration
NIC	network interface card
NIST	National Institute of Standards and Technology
NORM	NACK–Oriented Reliable Multicast
NPD	NASA Policy Directives
NPG	NASA Policy Guideline
NRL	Naval Research Laboratory
NSA	National Security Agency
NSTISSP	National Security Telecommunications and Information Systems Security Policy
NTP	Network Time Protocol
OMB	Office of Management and Budget
OS	Operating System
OSI	Open Systems Interconnection
PDA	Personal Data Assistant
PDD	Presidential Decision Directives
PGP	Pretty Good Privacy
PI	Principal Investigator
PROTEAN	PROTocol Engineering Advanced Networking
PUB	Public
QPSK	quadrature phase shift keying
R/S	Reed-Solomon
RF	radio frequency
RFC	request for comments
RMT	Reliable Multicast Transport
S/C	spacecraft
SAN	storage area network
SCID	spacecraft identifier
SCOS	spacecraft operating system
SCP	secure copy
SCPS	Space Communications Protocol Standards
SDSI	Simple Distributed Security Infrastructure
SER	Scientific and Engineering Research
SMTP	simple mail transfer protocol
SN	Space Network
SNTP	simple network transfer protocol
SONet	Synchronous Optical Network

SPKI	Simple Public Key Infrastructure
SPTR	South Pole TDRSS Relay
SSH	secure shell
SSL	Secure Sockets Layer
SSTL	Surrey Satellite Technology, Ltd.
STRV	Space Technology Research Vehicle
SW	software
TBD	to be determined
TBS	to be supplied
TCP	Transmission Control Protocol
TDM	time-division multiplexing
TDRS	Tracking and Data Relay Satellite
TDRSS	Tracking and Data Relay Satellite System
TLS	Transport Layer Security
TPC	Turbo Product Code
UDP	User Datagram Protocol
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
VC	virtual channel
VLAN	virtual local area network
VPN	virtual private network
VCID	virtual channel identifier
WAN	wide area network